DEVA004 PCI servo motion controller
CMM controller
stepper motion controller.

User's Manual

52 Woodside Business Park
Birkenhead
Wirral CH41 1EL
United Kingdom

Tel +44 (0)151 647 3222
Fax +44 (0)151 647 4511

Email: support@deva.co.uk

Web: www.deva.co.uk

All information of a technical nature and particulars of the product and its use are given by Deva Electronic Controls Ltd. in good faith. However, it is acknowledged that there may be errors and omissions in this manual. We shall not be liable for loss or damage whatsoever arising from the use of any information or particulars in, or any omissions from, this document

1 Overview 1
1.1 Product features 1
1.1.1 Controller features 1
1.1.2 Servo motion controller variants 1
1.1.3 CMM servo motion controller variants 2
1.1.4 Stepper motion controller variants 2
1.1.5 Motion controller usage 2
1.2 Typical applications 2
1.2.1 Automatic Co-ordinate measuring machine 2
1.2.2 Automatic test equipment 3
1.2.3 Special purpose machines 3
1.3 Support software 3
1.3.1 Windows XP/Win7-32/Win7-64 3
1.3.2 Axis tuner 4
1.4 Accessories 4
2 Installation and configuration 5
2.1 Software support CDROM 5
2.2 System requirements 5
2.3 Card installation 5
2.4 Windows XP/Win7-32/Win7-64 installation 6
2.5 Windows motion library simulation installation 6
3 Motion library usage 7
3.1 Functionality 7
3.2 Theory of operation 7
3.2.1 Servo control handler 8
3.2.1.1 Servo control loop 8
3.2.1.2 Model control 9
3.2.2 Move generation 10
3.2.3 Servo update system 10
3.3 Axis setup guide 11
3.3.1 Overview 11
3.3.1.1 Axis drive amplifier adjustment 11
3.3.1.2 Closed-loop position controller adjustment 11
3.3.2 Initial settings 12
3.3.3 Position feedback scaling 13
3.3.4 Open-loop axis drive response 13
3.3.5 Analog output scaling and phasing 14
3.3.6 Closed-loop position control 15
3.3.7 Additional control terms 16
3.3.8 Axis protection terms 16
3.3.9 Summary 16
3.4 Error messages 19
4 Configuration files 24

4.1 IO system definitions 24
4.1.1 IO system types 25
4.1.2 IO configuration 26
4.2 System configuration file 27
4.2.1 SIMULATION 27
4.2.2 UPDATE 27
4.2.3 AUTOINT 27
4.2.4 PRBTMR 28
4.2.5 WDOGTMR 28
4.2.6 APPWATCHDOGTMR 28
4.2.7 DRIVEDISABLETMR 28
4.2.8 DIGITAL_IO_BOARD_TYPE and DIGITAL_IO_BASE 28
4.2.9 PROBE_OVERSHOOT_DISTANCE 29
4.2.10 PROBE_CLEARANCE_DISTANCE 29
4.2.11 PROBE_JOYSTICK_PROTECT_DISTANCE 29
4.2.12 PROBE_JOYSTICK_PROTECT_FEED 29
4.2.13 ESTOPIP 30
4.2.14 SUPPLY_ON_IP & SUPPLY_ON_ANY_IP 30
4.2.15 ESTOP_CHAIN_OK_IP & ESTOP_CHAIN_OK_ENABLE_IP 30
4.2.16 NO_MACHINE_FAULT_IP 31
4.2.17 MACHINE_ON_IP 31
4.2.18 SYSTEM_ACTION 31
4.2.18.1 System action id 32
4.2.18.2 System action source / destination parameters 32
4.2.18.3 System action example 33
4.2.19 POSN_CALC_TOLERANCE 34
4.3 Axis configuration file 34
4.3.1 NAME 34
4.3.2 TYPE 34
4.3.3 DRIVE_ENABLE_FLAGS 35
4.3.4 FEEDBACK_MONITOR_FLAGS 35
4.3.5 CHANNEL 36
4.3.6 PITCH and COUNT 36
4.3.7 AUXCHANNEL 37
4.3.8 AUXPITCH and AUXCOUNT 37
4.3.9 ANALOGCHANNEL 37
4.3.10 ANALOGCHANNEL_PITCH and ANALOGCHANNEL_COUNT 38
4.3.11 MAXVOLT 38
4.3.12 DRIVESCALE 38
4.3.13 DRIVEOFFSET 39
4.3.14 DBMVOLT 39
4.3.15 MAXSPEED 39
4.3.16 PREDATUM_MAXSPEED 39
4.3.17 MAXACCEL 39
4.3.18 MAXDECEL 40
4.3.19 MAXJERK 40
4.3.20 EMERGACCEL, EMERGDECEL and EMERGJERK 40
4.3.21 KV 40
4.3.22 KD 41
4.3.23 KI and LIMITKI 41

4.3.24 VP 41
4.3.25 VD 41
4.3.26 VI and LIMITVI 42
4.3.27 KS 42
4.3.28 VFF 42
4.3.29 AFF 43
4.3.30 ACTIVATE_CONTROL_DEADBAND, MAINTAIN_CONTROL_DEADBAND and CONTROL_DEADBAND_FLAGS 43
4.3.31 SRVERR and MINSRVERR 44
4.3.32 POSNBAND and POSNTIME 44
4.3.33 POSLIMIT and NEGLIMIT 44
4.3.34 DATUM and DATUMMODE 45
4.3.35 DATUMSWITCHIP, DATSWITCHSPEED and DATMARKSPEED 46
4.3.36 POSOVERTRAVELIP, NEGOVERTRAVELIP 46
4.3.37 OVERTRAVELIPMODE 46
4.3.38 SKEW_PAIRED_CHANNEL, SKEW_GAIN and SKEW_LIMIT 46
4.3.39 SIMUL 48

5 PCI motion control hardware 49

5.1 Functional description 49
5.1.1 Quadrature input 49
5.1.2 Marker input 49
5.1.3 Renishaw probe interface 49
5.1.3.1 Probe input PICSB 50
5.1.3.2 Probe input PICSA 50
5.1.4 Digital IO 50
5.1.5 Pulse Generator 50
5.1.6 Axis Compare 51
5.1.7 Timer 0 51
5.1.8 Timer 1 51
5.1.9 Time Stamper 51
5.1.10 Event System 51
5.1.10.1 Incremental board level block diagram 52
5.1.10.2 Channel level block diagram 53

5.2 Axis connection details 54
5.2.1 Axis signal descriptions 55
5.2.2 Renishaw probe input connections 56
5.2.3 Digital / Analog IO connections 57
5.2.3.1 Feature connector 57
5.2.3.2 IO1 Digital IO connector 58
5.2.3.3 IO2 Internal servo connector 59
5.2.4 Sync connector 60

5.3 Direct hardware programming 60

6 PCI Stepper control hardware 61

6.1 Functional description 61
6.1.1 Stepper outputs 61
6.1.2 Digital IO 61
6.1.3 Timer 0 61
6.1.4 Event System 61

6.1.4.1 Board level block diagram 62
6.1.4.2 Channel level block diagram 63
6.2 Axis connection details 64
6.2.1 Axis signal descriptions 65
6.2.2 Digital IO connections 66
6.2.2.1 Feature connector 66
6.2.2.2 IO1 Digital IO connector 67
6.2.2.3 IO2 Internal servo connector 68
6.2.3 Sync connector 69
6.3 Direct hardware programming 69

1 Overview

The DEVA004 4-axis motion controllers have been designed to enable simultaneous control of up to 16 servo axes using a PC based system. They can be used for a wide range of motion control applications, such as CNC machines, robotics and co-ordinate measuring machines.

The range includes support for both PCI and ISA bus interfaces. Both types share many common features, however the PCI variant has several additional facilities. There is also a CMM specific controller.

1.1 Product features

The cards are based upon Deva's own ASIC technology, and have the following features:

1.1.1 Controller features

- Four encoder inputs (auto sensing for differential or single-ended input)
- Four analog outputs, with 16-bit D/A resolution (16 bit speed register on stepper variants)
- Four opto-isolated servo enable outputs
- Opto-isolated watchdog output
- Marker input for each encoder channel
- Probe input, to capture all channels simultaneously
- +5v +/-12v power supplies available on axis connectors
- Opto-isolated 16 digital inputs and 16 digital outputs
- 16 TTL level digital IO
- Flexible event system
- Axis compare
- Pulse generator
- Time stamper

1.1.2 Servo motion controller variants

- Three axis board with digital IO interface (X, Y, Z, 16 inputs, 16 outputs)

- Four axis board with digital IO interface (X, Y, Z, S, 16 inputs, 16 outputs)

1.1.3 CMM servo motion controller variants

- Three axis board with Renishaw probe interface and digital IO interface (X, Y, Z, PICSA, PICSB, 16 inputs, 16 outputs)
- Four axis board with Renishaw probe interface and digital IO interface (X, Y, Z, S, PICSA, PICSB, 16 inputs, 16 outputs)

1.1.4 Stepper motion controller variants

- Three axis board with digital IO interface (X, Y, Z, 16 inputs, 16 outputs)
- Four axis board with digital IO interface (X, Y, Z, S, 16 inputs, 16 outputs)

1.1.5 Motion controller usage

The 4 analog outputs are designed for connection to servo amplifiers, to control axis movement. The 4 encoder inputs are designed for reading axis position. These two elements form the basis of a closed-loop motion control system, with the PC itself controlling the axis positions, velocities and accelerations in software.

Of course, the DEVA 004 can also be used for other applications involving the handling of analog and encoder signals by a PC, such as speed control and position measuring systems.

The DEVA 004 is supplied with a powerful motion library along with variety of drivers and demonstration programs. Whilst it is possible to access the cards functions directly by using low level device driver functions and hardware access, it is recommended that any software is written to take advantage of the facilities provided by the device driver and motion libraries.

1.2 Typical applications

The following examples are given to illustrate how the Deva 004 may be effectively used with a target application.

1.2.1 Automatic Co-ordinate measuring machine

The DEVA004 CMM controller provides all the required motion control and probing systems to implement a fully featured automatic co-ordinate measuring machine controller. CMM's are typically driven by small, high precision servo motors and rely on air-bearings to reduce friction. Linear scales often provide feedback with resolutions in the region of 1 micron. A touch trigger probe is normally used as the sensor and may be wired directly to the DEVA004 CMM controller that is also able to

directly control the probe led. The DEVA004 CMM controller implements a wide variety of probe related motion profiles and measurement cycles to optimise the performance from the machine. Features such as smooth motion profiles, plane rotation of circular moves, and 'S' curve velocity functions further enhance the machine performance.

1.2.2 Automatic test equipment

Automatic test systems often need to move components around during a test procedure or adjust components under test. The PC computer has become extremely popular in the automation of test systems and hence many interfaces have become available to provide connection to a wide variety of measuring / processing equipment. The DEVA004 expands this capability by provided a highly flexible method for integrating motion control elements into the test system. The DEVA004 is provided complete with a powerful motion control library, which is readily accessible from programming systems such as Visual C++ and Visual Basic. This allows the systems engineer to concentrate on integrating the motion elements into his application.

1.2.3 Special purpose machines

The PC computer is being increasingly used as the control system in special purpose equipment such as grinding, bending, forming and profiling machines. The PC display may be configured and designed to offer exactly the level of functionality / complexity that the application demands whilst the programming environments are powerful enough to meet the needs of machine control. The DEVA004 provides the motion control system needed to automate the machinery in a highly versatile and flexible manner and is equally suited to controlling a high precision grinding machine as to positioning the tooling on a pipe-bending machine. The DEVA004 offers the OEM an extremely cost effective solution to adding motion control to a PC system.

1.3 Support software

A variety of software drivers, libraries and utilities are provided with the DEVA004 card to enable software development to be performed within a number of operating systems and applications. Software support is an ongoing activity, if support for a particular application or operating system is not currently provided, please call the Deva office to determine its availability.

1.3.1 Windows XP/Win7-32/Win7-64

PCI cards are supported in Windows XP/Win7-32/Win7-64 by Windows driver model drivers (WDM). Please contact Deva for information regarding support for legacy ISA cards. The drivers all support the standard Deva motion library functions. For development and demonstration purposes, a Win32 dynamic link library (DLL) is also provided which allows all motion library operation to be simulated on a PC without a card fitted. Please refer to section 5 for more details on the motion library.

'C' and Visual Basic example programs illustrate use of the motion library functions whilst example System and Axis configuration files provide an example of a typical control system configuration.

1.3.2 Axis tuner

A powerful axis tuning utility is supplied which may be used to assist with the configuration and tuning of the servo axes in a system. This is useful to allow users to quickly verify that the DEVA004 is installed correctly and to test a system without having to write their own software.

The program features control panel and graphical display windows. The control panel is used to control the axes and make changes in system and axis configuration data whilst the graphical display window illustrates the effects of changes in configuration data on the axis performance.

The program automatically loads the axis configuration file (**AXIS.CFG**) from the windows directory and the user selects the axis to be tuned. Parameters relating to the feed back, motor performance, PID gains and servo protection monitoring may be edited and updated using the apply button. In addition, the user can optionally change the servo update rate, close the control loop and disable software over-travel limit protection.

A manual jog facility is provided to allow the user to jog the axis in either direction, zero the axis position, perform a datum cycle and reset servo errors. An automatic cycle may also be configured which repeatedly moves the axis at a preset speed between two user-defined positions. This feature is very useful when fine-tuning the drive system.

The control panel displays a variety of status data for the axis and features display meters that indicate how well the speed scaling and directional balance of an axis have been adjusted.

After performing an automatic move cycle, the graphical display window provides a display of both the model axis and real axis velocities achieved during the motion. The shape of the real axis velocity profile allows the user to determine the stability and assess the performance of the axis under test.

Step and speed response of the axis may be tested by using the force voltage button that sets the analog output to the specified value.

1.4 Accessories

The DEVA004 motion control interface is supplied with 'D' type connectors and matching shells for the cabling of the encoder inputs and the analog control outputs.

A variety of cables and connectors are available to allow connection to digital I/O and Renishaw probe interfaces. Please contact Deva with your requirements.

2 Installation and configuration

2.1 Software support CDROM

The DEVA004 motion control card is supplied with a software support CDROM containing support for all DEVA004 motion control cards along with support and information for many of Deva's other products. The CDROM includes the following items:

Windows XP/Win7-32/Win7-64 device drivers providing coherent motion control card hardware management.

Windows XP/Win7-32/Win7-64 DLLs and import libraries containing the motion control library functions.

'C' language library routines and header files which provide constant and function definitions for access to the motion control library.

Example 'C', Visual 'C' and Visual Basic programs illustrating use of the motion control library.

A Visual Basic module provides constant and function definitions for access to the motion control library.

Example Visual Basic programs illustrating use of the motion control library.

Demonstration Axis Tuner Program for Windows.

Motion control configuration files (**SYSTEM.CFG**, **AXIS.CFG**)

2.2 System requirements

The device driver library functions and the demonstration software require an IBM PC compatible (Pentium or above recommended) with one spare 5V, 32 bit PCI slot.

2.3 Card installation

The DEVA004 requires 4k bytes of memory. This space is automatically assigned by the Plug and Play manager (OS/BIOS) when the computer is rebooted with the card installed. The card is installed by inserting it into any available 5V or 3.3V, 32 bit PCI/PCIX slot.

2.4 Windows XP/Win7-32/Win7-64 installation

During the first re-boot after the card has been installed windows will indicate that a new PCI device has been found and will start the standard driver installation procedure. If this does not occur it is possible to initiate this process manually via the 'add new hardware' icon in the control panel or via the windows device manager.

Follow the instructions and when requested select 'have disk' and then browse to the directory on the installation CDROM containing the Deva004.inf file.

For example, select:

\PC interface products\Deva004\Issue3.x\Drivers\\Deva004.inf

Click ok and follow instructions to complete the installation.

The installation may be tested using the supplied Axis Tuner program that may be found on the CD in the utils\win32 directory.

2.5 Windows motion library simulation installation

For demonstration and development purposes, it is possible to simulate operation of a servo control system without a Deva004 card or any associated hardware. To use the Windows motion library in simulation only a single DLL is required:

CNTRL32. DLL - 32 bit motion library DLL

This CNTRL32.DLL should be copied from the relevant drivers directory on the distribution CDROM i.e. WinXP or Win2000. This file may be placed either in the directory with the users application or, more conventionally, in the Windows system directory.

To run the utility programs supplied on the disk the **SYSTEM.CFG**, **AXIS.CFG** and **MESSAGE.TXT** files must also be copied onto the system. These files should be placed in the Windows directory.

The CNTRL32.DLL operates in simulation when the **SIMULATION** parameter is set to '1' in the SYSTEM.CFG file i.e.

SIMULATION=1

Please see the **System configuration file** section for more detailed information on use of this parameter.

3 Motion library usage

3.1 Functionality

The following table illustrates the functionality available from the motion control software.

Feature	
Maximum number of boards	4
Maximum number of axes	16
Spare analog outputs available	√
Configurable auxiliary feedback channel	√
Digital I/O support	√
Integrated datum cycle	√
Integrated joystick support	√
Full PID with VFF	√
Speed loop with PID and AFF for torque mode	√
Split axis with skew gain for gantry type axes	√
Shared axes with common motor or amplifier	√
Servo error protection	√
Hardware watchdog protection	√
Software over-travel protection	√
Hardware over-travel limits	√
Configurable hardware over-travel mode	√
Configurable acceleration	√
Configurable deceleration	√
Configurable jerk	√
Configurable axis drive deadband voltage	√
Axis simulation capability	√
Customise operation with 'system action' facility	√
Maximum parallel motions	16
Move queue	√
Maximum move queue items	64
Event queue	√
Linear interpolation	√
Circular 2D, 3D and helical 3D interpolation	√
Continuous motions with non-zero final feed	√
Integrated probe point and protection cycles	√
Fifo data logging buffer length	4000
Filtered axis velocity available	√
Motion library simulation facility	√
Configurable control deadband	√
Hardware event system	√
Application watchdog	√
Virtual axes	√
Slaving of real axes to virtual and real axes	√

3.2 Theory of operation

The motion control library is designed to provide the facilities necessary for control of axis motions from within a user application. A set of command functions are employed that allow a user to configure the control system, request axis motions and control

motions in progress. These command functions interface to the system servo control handler that is responsible for maintaining closed loop axis position control.

Under Windows, the user installs the appropriate device driver for the operating system as described in the installation section. This driver integrates the servo control handler with functions required for hardware access. The motion library command functions are contained in a DLL that communicates with the device driver. The user must call library routines in the 32 bit 'cntrl32.dll' to access the functions described in this manual from 32 bit Windows applications written in Microsoft Visual Basic, Microsoft C++, Borland Delphi, etc.

Various 'C' header files and Visual Basic modules are provided to enable the developer to interface to the motion library.

3.2.1 Servo control handler

The servo control handler employs a conventional following error based system to control the position of axes. Following error is defined as the difference between the model position of an axis and the actual position of that axis. The control loop drives the axes to maintain as small a following error as possible. The servo controller is therefore able to create axis motions by controlling the model position of each axis. For accurate control of an axis, it is important that the servo control handler updates the servo control loop and model positions at regular intervals.

In addition to the position control loop the servo controller optionally includes a velocity loop for axis drives that operate in torque or current mode.

3.2.1.1 Servo control loop

The servo control loop continually compares the actual and model position of each real axis and determines the required speed for the axis. The difference between the actual and model position of an axis is defined as following error.

The speed demanded by the servo controller is related to following error by proportional gain 'Kv' according to the following expression:

$$Speed_{demand} = \left(Position_{\mathrm{mod}\,el} - Position_{actual}\right) \times Kv$$

The derivative gain 'Kd' and integral gain 'Ki' also affect the speed demand and may be employed where appropriate for the system.

Typically, the axis drive amplifier will provide closed loop speed control of the axis and therefore speed demand is proportional to the voltage reference applied to the axis drive. Parameters are provided that permit the speed demand to be scaled to the analog voltage reference and these should adjusted according to the individual axis drive electronics and mechanical arrangements.

As the analog voltage applied to the axis drive is proportional to following error, the axis must move if the model and actual positions of an axis are different. For example, if the following error is positive the axis will move in a positive direction until the

following error is zero. Therefore, the action of the control loop is to control the speed of an axis so that the difference between its model and actual positions is as small as possible.

The response of the control loop is determined by the value of proportional gain 'KV'. For a given following error, the higher the value of 'KV', the larger the analog voltage that will be applied to the axis drive. If the 'KV' is too high, instability or oscillation of the axis may result. Conversely, too low a value of 'KV' will produce a sluggish response from the axis. Typical values of 'KV' range from 10 /s to 200 /s.

If the model position of an axis is continuously incremented by a fixed amount at a regular interval, the model position may be considered to be moving with constant velocity. The control loop reacts by moving the real axis towards its incremented model position. Assuming the control loop is correctly tuned, steady state conditions arise after a short time and the following error becomes stable. Under these conditions, the stable following error and proportional gain combine to produce a stable analog voltage to the axis drive. Hence, constant axis speed is maintained and the axis position lags the behind its model position by the amount of the following error.

In some cases, the axis drive amplifier operates in torque or current mode and the servo control loop must provide the closed loop speed control of the axis. In this configuration, the voltage reference applied to the axis drive is proportional to torque or current and thus acceleration. The acceleration demanded by the servo controller is related to the error in speed by the speed loop proportional gain 'Vp' according to the following expression:

$$Accel_{demand} = \left(Speed_{demand} - Speed_{actual}\right) \times Vp$$

It is to be expected that, for stability of a correctly scaled system, the position loop gain 'Kv' is lower than the speed loop gain 'Vp'. The derivative gain 'Vd' and integral gain 'Vi' also affect the acceleration demand and may be employed where appropriate for the system.

3.2.1.2 Model control

The servo control loop continually adjusts the analog voltage applied to an axis drive so that the axis moves towards its model position. Therefore, the servo controller is able to create axis motions by controlling the model position of each real axis.

A motion is generated by incrementing or decrementing the model position of an axis at regular intervals. For example, to move an axis 1000 units in 1 second, the model position could be incremented by 1 unit every 1ms or 2 units every 2ms depending on the update rate of the servo control handler. For a given update interval, the larger the increment applied to the model position of an axis, the larger the following error and hence the faster the axis will move towards its new model position. The servo control handler is therefore able to define the speed and position of each axis.

A master, model axis is created by functions within the motion library in response to a request for an axis motion. This model axis may be considered as a simulated axis whose required motion is the vector addition of all axis movements defined in a

particular motion. The servo control handler calculates a new position for the master model axis at every update interval. The model axis effectively performs a motion at the requested speed using appropriate acceleration and deceleration ramps.

The servo control handler derives a model position for each real axis included in a motion from the position of the master model axis. Therefore, as the model axis performs its motion, the servo control loop will move each axis towards its updated model position. Complex, interpolated axis motions may be created by defining the relationship between the position of the master model axis and the model position of each real axis.

Multiple model axes may be managed by the system. This allows combinations of independent and interpolated axis motions to be performed simultaneously. For example, two orthogonal axes may perform continuous, interpolated circular motions whilst a third axis moves only in response to an asynchronous event.

3.2.2 Move generation

A powerful yet simple set of functions is provided which allows a wide variety of motions to be defined and executed. Motions involving different axes may be executed serially or in parallel. A system of move queues allows the user to create and put moves into a buffer. This ensures that a following move may be prepared and made ready in the buffer before the current move is completed. As each move may be started at the speed that the previous move completed, this permits moves to follow each other smoothly and without stopping.

3.2.3 Servo update system

To accurately control axis motion the servo control loop and model control functions performed by the servo control handler must be executed at regular intervals. Typical update intervals are between 1ms and 5ms with the interval selected depending on the dynamic performance required of the axes. The higher the update rate, the quicker the servo control loop will react to changes in the process to keep an axis on profile. If the user selects a particularly high update rate a slow down in the performance of the PC may be noted.

The DEVA004 hardware incorporates a timer driven interrupt controller that may be configured to produce a hardware interrupt at a programmable rate. The system configuration defined by the user application determines the required interrupt rate. Functions within the motion library automatically set the interrupt rate to this value. An interrupt handler is installed which services the interrupt before executing the servo control loop and model control functions of the servo control handler.

3.3 Axis setup guide

3.3.1 Overview

As the size, speed, resolution and accuracy of servo drive and feedback systems vary greatly it is necessary to adjust and configure the control system components to account for the specific electrical and mechanical arrangements employed.

3.3.1.1 Axis drive amplifier adjustment

Although the Deva004 closed-loop position controller may be configured to operate with an axis drive amplifier and motor operating in 'acceleration-torque-current' mode, the description in this section assumes that the axis drive amplifier and motor operate in 'speed' mode. This implies that the axis drive amplifier is actually a closed-loop speed controller using feedback from the motor such as a tachometer, encoder, resolver or back-EMF to regulate the speed of the axis. It is therefore essential that the axis drive amplifier and motor are setup and adjusted according to the manufacturers recommendations so that motor speed produced is proportional to the analog speed reference voltage applied by the motion control card.

Furthermore, before the closed-loop position controller can be used, the various settings and gains of the axis drive amplifier closed-loop speed controller must be adjusted so that the response of the axis to a step-input speed demand is acceptable. This adjustment was traditionally performed before connecting a motion controller using a 'battery box' and oscilloscope. A 'battery box' is simply a battery wired to a switch and potentiometer that could be used to supply step-input speed demands of varying amplitudes to the axis drive. The oscilloscope is connected to a motor tachometer in order to provide visualisation of the motor response so that the axis drive gains and settings could be adjusted to provide the required response.

However, in place of the 'battery box' and oscilloscope the Axis Tuner program supplied on the distribution CD may be used to enable the axis drive and to 'force' step-input speed demand voltages to be applied to the axis drive amplifier. The response of the axis is displayed on a graph so that the axis drive amplifier closed-loop speed controller can be adjusted.

Although not recommended, it should be noted that 'linear' amplifiers offering only open-loop speed control may be employed in certain circumstances where the user is able to establish that the resulting limited level of system performance is adequate.

3.3.1.2 Closed-loop position controller adjustment

Once the axis drive amplifier and motor have been adjusted to provide a satisfactory response to a step-input speed demand, the Deva004 closed-loop position controller may be configured. Each axis is configured by altering a set of numerical parameters in the **AXIS.CFG** file. The contents of the file and meaning of each of the parameters are described in section 4. The aim of tuning the closed-loop position controller parameters is to arrive at a set of parameters that will provide the axis performance required by the system designer whether controlling large high-inertia, low speed, low

gain machine tools or light-weight, low-inertia, friction-free co-ordinate measuring machines.

For the Deva004 closed-loop position controller to operate correctly the position feedback parameters are adjusted so that system measures the correct distance and in the right direction. Default values for acceleration and servo error protection bands are defined. Speed scaling parameters define the relationship between analog speed reference voltage applied by the motion control card to the axis drive amplifier and the physical speed of the axis achieved. This scaling is important if the units of the PID terms are also to be correctly scaled and meaningful. The PID gains are themselves optimised to produce a stable axis response whether for 'stiff' high-gain or 'sluggish' low-gain systems.

The following section describes a typical procedure that may be employed to produce a successful set of axis parameters. It is assumed that the Deva004 and driver software have been successfully installed and that an application is available which can call for axis motions, display encoder position, axis speed and axis lag / following error. It is suggested that the Axis Tuner program supplied on the distribution CD is used for this purpose. The Axis Tuner program allows the user to enter and change values of the axis configuration parameters dynamically and to view the effects of these changes on motions. Parameters are updated by pressing the 'Apply' button Once satisfactory parameter values have been determined, these values may then be manually copied into the **AXIS.CFG** file or exported to a separate file as required.

3.3.2 Initial settings

When setting up a system for the first time it is recommended to begin configuration with the following set of parameters. These settings have the effect of disabling all advanced functions to prevent them interfering with the basic set-up procedure.

Parameter	Value	Comment
NAME	X	Depends on axis
CHANNEL	0	Depends on axis
PITCH	1.0	1 unit = 1 counts
COUNT	1	(This is equal to one step in stepper systems)
MAXVOLT	9000	Or less if your drive cannot accept 9 volts
MAXSPEED	100	The axis travels 100 units/s for 9 volts
MAXACCEL	100	The axis takes 1 second to reach full speed
KV	10	
KD	0	
KI	0	
POSNBAND	1000.0	Set this large so it does not interfere
POSNTIME	1000.0	Set this large so it does not interfere
SERVOERR	1000.0	Set this large so it does not interfere
MINSRVERR	1000.0	Set this large so it does not interfere
SIMUL	0	Turn simulation off

All other parameters should be removed from the entry for the axis being configured. All other axes should be set to simulation (**SIMUL**=1) to avoid confusion.

3.3.3 Position feedback scaling

If the relationship between encoder counts and user units is known then the values of **PITCH** and **COUNT** can be set directly with reference to section 4. Either parameter may be set to a negative value to reverse the count direction so that it is consistent with the machine axes. For example if a 1um resolution linear scale was found to count negative when the axis moved positive, then the user could set either **PITCH=-1** and **COUNT=1000** or **PITCH=1** and **COUNT=-1000**.

If the above relationship is not known, it will be necessary to determine the correct values by experiment. With power removed from the motor drive system or with the servo loop open move the encoder over a set distance 'D' units and note the change in the axis position display.

Since **PITCH** and **COUNT** were previously set to 1, the display change 'C' is equivalent to the number of encoder counts moved. Now set **PITCH**=D and **COUNT**=C and repeat the experiment. The display should now read in true units.

When using a stepper control card the value of **COUNT** represents one step of the stepper system and **PITCH** how far the axis moves for one step.

N.B. Before proceeding please ensure that the axis position changes in both the required units and in the required direction for the machine.

3.3.4 Open-loop axis drive response

At this stage, there is no servo error protection of the axis, so take steps to protect the axis as necessary. If possible, ensure the drive is not mechanically connected to any equipment and is hence free to turn.

The 'force' box on AxisTuner may be used to enable the axis drive and force step-input speed demand voltages to the axis drive amplifier. This is open loop control and allows adjustment of the amplifier closed-loop speed controller without the effect of the close loop position control. The user enters the voltage to be applied in units of mVolts and uses the +/- buttons to apply the specified voltage to the axis drive amplifier. For example, entering a value of 1000 and pressing the '-' button will cause –1000mV to be applied to the axis drive amplifier as long as the button is held down and the motor should be seen to move at an appropriate speed.

The axis drive amplifier zero offset may be adjusted whilst applying a value of 0mV and pressing either of the +/- buttons. The correct zero offset is achieved when there is no axis motion detected with 0mV applied.

With a nominal value of 1000mV applied, the axis drive motor should move backwards and forwards at the same, constant speed as the +/- buttons are pressed in turn. The Axis Tuner displays a graph of axis speed against time illustrating the response of the axis. The amplifier closed-loop speed controller gains and parameters should be adjusted so that the displayed response is acceptable in both directions and there is no observed instability. An optimally adjusted drive might display perhaps one overshoot and one undershoot in response to a step-input speed demand.

To make best use of the dynamic range of the Deva004 closed-loop position controller, the axis drive amplifier speed scaling should be adjusted so that the maximum working speed required of the axis is achieved with around 9000mV applied to the axis drive. It is recognised that this may not always be possible, in which case a lower value may be employed. Gradually increase the forced voltage from 1000mV up to maximum (nominally 9000mV) checking that the axis drive response is acceptable and adjusting gains and settings if necessary. Adjust the axis drive speed scale until the 'speed' display on Axis Tuner indicates the axis is moving at the maximum working speed when maximum (nominally 9000mV) volts are applied.

For axes with restricted travel, it may not be possible to safely work up to the maximum working speed with this method. If this is the case, these procedures may be performed up to, for example, half or quarter speed. The axis drive speed scale should be adjusted so that the 'speed' display on Axis Tuner indicates the axis is moving at half or quarter speed as appropriate. Further adjustment of the axis drive may then be performed once the axis is under closed-loop position control of the Deva004.

Confirm that the relationship between speed demand voltage and axis speed is linear across the working speed range by applying a range of forced voltages and noting the speed that the axis travels at each voltage. The ratio of speed/voltage should be the same for all combinations.

Finally, re-check the axis drive amplifier zero offset is adjusted correctly by applying a value of 0mV, pressing either of the +/- buttons and making sure the axis does not move.

N.B. This step must be completed before attempting to close the position loop.

3.3.5 Analog output scaling and phasing

In many cases the velocity travelled by the axis for a given analog input to the servo drive is known (or can be estimated). If this is the case set the **MAXSPEED** parameter to the speed the axis will travel if **MAXVOLT** (mV) are present at the drive input. A negative value of **MAXVOLT** **must** be used if the axis position counts in a negative direction when a positive voltage is applied to the axis drive amplifier.

Alternatively, the Axis Tuner 'force' facility can be used to derive values for the **MAXVOLT** and **MAXSPEED** parameters. This is simply achieved by forcing the required maximum speed demand reference voltage to the axis drive (nominally 9000mV) and noting the speed of the resulting axis motion from the AxisTuner 'speed' display. If the axis moves in a positive direction when a positive voltage is applied, then set **MAXVOLT** to the applied voltage and **MAXSPEED** to the speed observed. However, if it is noted that the axis position counts in a negative direction when a positive voltage is applied, then a negative value of **MAXVOLT** **must** be used.

N.B. Always double check that the axis is phased correctly i.e. the axis moves in a positive direction at **MAXSPEED** when a value of **MAXVOLT** (taking into account the sign of this parameter) is applied to the axis drive amplifier speed demand. The axis may runaway out of control if the phasing is incorrect.

3.3.6 Closed-loop position control

Before attempting to close the position loop it is important to check the following:

The axis drive amplifier is operating correctly and the speed loop adjusted.
The position feedback device is scaled to count the correct units and in the correct direction.
The analog output is scaled and phased so that the axis moves in the correct direction at the correct speed.

If possible, ensure the drive is not mechanically connected to any equipment and is hence free to turn. Otherwise, consider reducing the **MINSRVERR** parameter to a lower value, for example 10 units, to limit the distance the axis will travel if it runs off out of control. For good measure set **MAXACCEL** to the same figure as **MAXSPEED** which gives a time of 1 second to reach a speed of the **MAXSPEED** parameter.

If using the Axis Tuner program, pressing the 'close loop' button will enable the axis drive output from the Deva004 motion control card and enable closed-loop position control for the axis. The axis should remain stationary, if the axis moves more than a small amount, then release the 'close loop' button and investigate before proceeding.

Attempt to jog the drive at a constant speed 'V' units/s. The drive should accelerate and then hold a steady speed. Note the speed and lag of the axis. If the drive sets off in the wrong direction, it is probable that the position feedback or analog output speed are not phased or scaled correctly. If the machine moves but displays instability, then the value of **KV** may be too high and should be reduced.

To check that the drive is under control, attempt to jog at a speed of 'V' / 2 units. Note the speed and lag. Both should be approximately half of the previous readings. If this is not the case then re-check the **MAXSPEED** and **MAXVOLT** parameters.

Once the drive is determined to be under control it is possible to check or refine the values of the parameters **MAXVOLT** and **MAXSPEED**. The lag recorded above should equal the velocity of the axis divided by **KV**. Since **KV** is previously set to 10, the lag should equal one tenth of the velocity or 10 units for 100 units/s. If this is not the case then the value of **MAXSPEED** is not correct and may be corrected using the following formula:

New **MAXSPEED** = old **MAXSPEED** * lag * V / **KV**

Once reasonable figures for **MAXSPEED** and **MAXVOLT** have been found it is often preferable to choose round figures for these parameters and fine tune the drive using the speed gain potentiometer. The speed 'scale' meter on Axis Tuner should indicate green with the needle in the centre when these parameters are correct and the axis drive speed gain potentiometer adjusted. These figures can then be simply copied to like axes on the same machine or a batch of machines.

Once the axis is moving at the requested speed with the correct lag the **KV**, **MAXACCEL, MAXDECCEL and MAXJERK** parameters may be adjusted to suit the system requirements. If the value of proportional gain **KV** is too low, the axis is 'soft' and unable to hold position whilst increasing **KV** 'stiffens' the axis and improves the

position holding. However, too high a value of **KV** could lead to instability. For machines such as machine tools and co-ordinate measuring machines, a value of **KV** from 10 to 30 is typical whilst smaller, more dynamic axes may achieve much higher values.

3.3.7 Additional control terms

On systems with unusual characteristics or requirements such as effects of gravity or low friction air bearings it may prove necessary to invoke additional control terms such as **KD, KI** and **VFF**.

If the machine displays traces of instability with low proportional gain **KV** values, consider using **KD** (derivative gain) to stabilise the axis and permit use of a higher proportional gain **KV** than would otherwise be practical.

The integral gain term **KI** is only activated when the axis is stationary and is used in combination with the **LIMITKI** parameter to improve the steady state performance of the control system in the presence of disturbances such as friction, offset drift, and gravity.

For systems that operate in 'acceleration-torque-current' mode, the **VP, VD** and **VI** parameters may be used to configure a speed loop within the servo controller.

Please refer to section 4 for suggestions as to their usage.

3.3.8 Axis protection terms

Once the axis is performing satisfactorily, it is possible to bring into operation the servo protection and axis limit parameters, **POSNBAND, POSNTIME, SERVOERR, MINSRVERR, POSLIMIT, NEGLIMIT.**

Please refer to section 4 for suggestions as to their usage.

3.3.9 Summary

Axis drive amplifier operating in 'speed' mode

Until the position loop is closed, there is no servo error protection of the axis, so take steps to protect the axis as necessary and disconnect from the mechanical load if possible. The following steps are performed using the Axis Tuner utility:

1. Ensure that the axis drive is operating in 'speed' mode.

2. Set **PITCH** and **COUNT** parameters so that the axis position feedback device is counting in the correct units and in the correct direction.

3. Force a small voltage (nominally +/-1000mV) and ensure the axis moves backwards and forwards at constant speed as the +/ force buttons are pressed.

If any instability is noted, adjust the axis drive amplifier speed loop according to the manufacturer's instructions to produce a stable response.

4. Force 0mV out and adjust the axis drive amplifier zero offset so that there is no creep of the axis.

5. Force maximum speed demand (nominally 9000mV) to the axis drive amplifier and adjust the axis drive amplifier speed scale so that the maximum working speed required of the axis is achieved.

6. Optimise the amplifier gains so that the response is acceptable (maybe one overshoot and one undershoot).

7. Confirm that the axis speed changes linearly and in proportion to a range of forced voltages in both +/- directions.

8. Recheck and adjust the axis drive amplifier zero offset if necessary.

9. Set **MAXSPEED** to the maximum working speed required of the axis and **MAXVOLT** to the maximum speed demand (nominally 9000mV). If the axis moves in a negative direction when a positive voltage is applied to the axis drive, then the value of **MAXVOLT** **must** be negative.

10. Consider setting the **MAXACCEL** and **MAXDECEL** parameters to the same as **MAXSPEED**, as this will give 1 second ramp to full speed as a starting point.

11. Set **KV**=10, **KD**=0, **KI**=0, press the 'Apply' button and press the 'close loop' axis button, the axis should now be live and under control. If the axis runs away or a servo error is generated, check the polarity of the **MAXVOLT** and **PITCH** parameters and adjust as necessary or review the steps performed earlier.

12. Use the Auto single/cycle buttons to cyclically move the machine between two positions. Axis Tuner's 'scale' and 'offset' meter needles should be roughly central during the constant velocity part of the motion and indicating green. Otherwise, review the steps performed earlier.

13. Fine-tune the axis drive amplifier speed scale so that the Axis Tuner 'scale' meter is central across a range of speeds. Alternatively, adjust the value of **MAXVOLT** to achieve this.

14. Fine-tune the axis drive amplifier zero offset so that the' lag ' or following-error displayed by Axis Tuner is as close to zero as possible. The value may flicker a small amount.

15. Most machines should be reasonably stable with a proportional gain **KV**=10, but if the machine displays traces of instability, consider using **KD** (derivative gain) to stabilise the axis. Only use small values of **KD**, maybe increments of 0.1 at a time. Too high a value of **KD** will make the axis noisy.

16. Attempt to increase **KV**, noting any effects on the stability of the axis. For machines such as machine tools and co-ordinate measuring machines, a value

of **KV** from 10 to 30 is typical whilst smaller, more dynamic axes may achieve much higher values.

17. When the axis is moving acceptably, enter the required parameter values into the axis.cfg file and press the 'Load config' button to reload the edited axis.cfg file. Recheck the axis motion. Alternatively, use the export button.

3.4 Error messages

Number	Message	Description
0	Ok	No error
-1	Error opening device driver	A call to open_control has failed to open the hardware device driver. Check installation of the motion control device driver.
-2	Error device driver access	There is an error accessing the MS-DOS device driver.
-3	Error device driver command ptr	An invalid or null command pointer has been returned by the MS-DOS device driver.
-4	Too many configured axes	The axis configuration file has more axes than are available for the configured/installed motion control cards. This error is no longer implemented.
-5	Servo tsr installed	This error is no longer implemented.
-6	Update rate out of range	This error is no longer implemented.
-7	Interrupts not detected	The call to open_control has failed as there are no hardware interrupts being serviced by the device driver.
-8	config file not opened	A request to open or load the axis or system configuration file has failed. Check the path and filename of the configuration file.
-9	Illegal pitch config	The PITCH parameter in the axis configuration file has been set to the illegal value of 0.0.
-10	Illegal count config	The COUNT parameter in the axis configuration file has been set to the illegal value of 0.0.
-11	Illegal maxvolt config	The MAX_VOLT parameter in the axis configuration file has been set to the illegal value of 0.0.
-12	Illegal maxvely config	The MAX_VELY parameter in the axis configuration file has been set to the illegal value of 0.0.
-13	Illegal config parameter	A parameter in the axis configuration or system configuration file has not been recognised or the parameter index used to read/write a config value is invalid. Check the spelling of the parameter name nmemonics or the parameter index used.
-14	Axis channel not configured	A call to register_axis has failed to locate an axis configured with the specified hardware channel. Check the axis configuration file for allocated axis channels.
-15	Non-existant axis id	A call to a function has been made with an invalid axis id. Check that register_axis has returned a non-zero axis id value.
-16	Axis control ptr required	A function has been called that requires a REAL_AXIS_CONTROL structure to be allocated via the register_axis function. Allocate a control structure before using the relevant function.
-17	Axis status ptr required	A function has been called that requires a REAL_AXIS_STATUS structure to be allocated via the register_axis function. Allocate a status structure before using the relevant function.

Number	Message	Description
-18	No free models	There are no free models available for the requested motion. This error should only occur if multiple parallel motions have been requested for every axis in the system and a new move is requested whilst the axes are still in motion. Check that motions are completed before starting a new move.
-19	Circle without cos axis	A call to start_circle has been made without an axis defined as CIRCULAR_COS.
-20	Circle without sin axis	A call to start_circle has been made without an axis defined as CIRCULAR_SIN.
-21	Non-existant move id	A call to a function has been made with an invalid model id. Check that the move function has returned a non-zero model id value.
-22	Null data ptr	Check that the handle to the data structure to read/write is not null.
-23	Not able to alloc dos memory	The function has been unable to allocate enough memory for the requested operation.
-24	Cos axis already defined	A call to start_circle has been made with two or more axes defined as CIRCULAR_COS.
-25	Sin axis already defined	A call to start_circle has been made with two or more axes defined as CIRCULAR_SIN.
-26	Servo error stall	The axis in motion has fallen behind the 'ideal' axis position by more than the distance specified by the SRVERR parameter in the axis configuration file. This indicates either the axis has physically stalled due to drive/mechanical problems, is incorrectly configured or that the SRVERR parameter is set to too small a value for the performance of the physical axis. Also, check that the axis position feedback is operational.
-27	Servo error runaway	The axis in motion has overtaken the 'ideal' axis position by more than the distance specified by the SRVERR parameter in the axis configuration file. This indicates that either the axis has runaway out of control due to drive/mechanical problems, is incorrectly configured or that the SRVERR parameter is set to too small a value for the performance of the physical axis.
-28	Servo error standstill	The stationary axis has unexpectedly moved from its standstill position by more than the amount specified by the MINSRVERR parameter in the axis configuration file. This problem may be caused by incorrect offset adjustment of the axis drive or external mechanical forces.
-29	Servo error position tol	On completion of a motion, the axis has failed to reach target position within the band specfied by the POSNBAND parameter in the axis configuration file after the timeout period specified by POSNTIME. This problem may be caused by incorrect offset adustment of the axis drive or external mechanical forces.
-30	Pos software overtravel	The axis has reached the maximum position specified by the POSLIMIT parameter in the axis configuration file.
-31	Neg software overtravel	The axis has reached the maximum position specified by the NEGLIMIT parameter in the axis configuration file.

Number	Message	Description
-32	Pos overtravel switch	The axis has reached the positive overtravel switch either configured by the POSOVERTRAVELIP or set via the pos_overtravel_switch member of the REAL_AXIS_CONTROL structure.
-33	Neg overtravel switch	The axis has reached the positive overtravel switch either configured by the NEGOVERTRAVELIP or set via the neg_overtravel_switch member of the REAL_AXIS_CONTROL structure.
-34	Illegal vff value	The VFF parameter in the axis configuration file has been set to an illegal value outside the range 0.0 to 1.0.
-35	Illegal error number	This error is not currently implemented.
-36	No datum cycle configured	The user has requested a datum cycle to be performed for an axis that has no datum cycle configured in the axis configuration file.
-37	Axis not at target	An axis is in motion and it is not possible to start the new move.
-38	Queue blocked for moves	The move queue is currently blocked for adding further moves. The queue will become unblocked in as the move sequence progresses.
-39	Move queue full	The move queue is currently full. There will be space for further moves once a move that is currently stored has been executed.
-40	Move queue empty	There are no moves in the queue to be read.
-42	Move queue flushed and not yet complete	The move queue has been flushed and it is not possible to add further moves until the move in progress has completed.
-41	Invalid parameter	An invalid parameter has been passed to a function
-43	Circle geometry error	The system is unable to create a circular motion from the programmed data.
-44	Axis in error condition	The function has failed because of a servo error condition on one of the axes involved.
-45	Invalid move mode	A motion has been requested with invalid MOVE_MODE_TYPES.
-46	Invalid move type	A motion has been requested with invalid MOVE_PARAM_TYPES.
-47	Axis not at start position	An axis is not at the required start position for a move read from the move queue by the servo controller.
-48	Axis start and target mismatch	The start position of an axis must correspond to the last programmed target position for an axis.
-49	Invalid 180 arc	The start and end point of a CIRCULAR_MOVE_3D_TYPE motion are at 180 degrees. The system cannot resolve the direction of motion required.
-51	Probe not reset	A move with a PROBE_MODE selected has been started with the probe deflected and not seated.
-52	Probe protection invoked	The probe has triggered during a move with the PROBE_PROTECT mode selected.
-53	Illegal probe mode	The programmed probe mode is not recognised or is not compatible with the motion.
-54	Illegal system action id	The specified system action id is not recognised.
-55	Illegal motion id	The specified motion id has not been recognised.

Number	Message	Description
-56	No free motions	The user has exceeded the number of motion ids that may be allocated for the system.
-57	Not enough free queue items	There are not enough free queue items left to create the move queue size requested.
-58	Probe not triggered	The probe did not trigger during a motion with PROBE_POINT mode selected.
-59	Report axis failure	The index supplied to the report_axis function is invalid.
-60	Joystick driver not detected	The system was unable to find a joystick driver.
-61	Joystick device not detected	The system was unable to find a joystick device.
-62	Multimedia timer not enabled	The system has no free multi-media timers available
-63	Invalid IO system	The specified IO system is invalid
-64	No inputs available	There are no inputs available in the IO system
-65	No outputs available	There are no outputs available in the IO system
-66	Probe protection invoked during backoff	The probe has re-triggered whilst backing off from a point during a motion with the PROBE_POINT_BACKOFF attribute.
-67	Emergency stop input activated	The **ESTOPIP** input has been set to a logic level 0 to indicate that the system is in an emergency stop condition.
-68	Move not permitted on measurement axis	The axis is configured as a measurement axis and may not be included in the motion.
-69	Watchdog occurred	The on-board timer watchdog has tripped, the card has not been serviced for longer than the preset watchdog period.
-70	Motion aborted	A configured system action has been triggered and all moves in progress aborted.
-71	Emergency stop	A configured system action has been triggered the controller has entered an emergency stop condition.
-72	Skew error	The pair of axes configured as split axes with skew gain have 'skewed' by more than the amount specified by the **SKEW_LIMIT** parameter.
-73	Feedback disconnected	The feedback device appears to be disconnected as A, B and M are not in a valid combination.
-74	Feedback quadrature error	The A and B signals have jumped to an invalid quadrature state,
-75	Feedback parity error	A parity error has been detected on the SSI feedback data.
-76..-85	Emergency stop system action id 1..10	A user configured emergency stop system action defined with an id from 1..10 has occurred.
-86	Application watchdog occurred	A 3rd party application has enabled the application watchdog but failed to call the **kick_application_watchdog** function within the period defined by the **APPWATCHDOGTMR** parameter in the system.cfg file.
-87	Mismatched drivers	The system has detected that components of the motion control library driver stack are not version or interface compatible. It may be necessary to check the versions of the motion control library driver files to establish the cause of the problem. Alternatively, re-install a known good set of drivers from the installation CD.

Number	Message	Description
-88	No free axes	There are no free axes for the requested operation.
-89	Connected axis error	There is an error on another axis to which the axis displaying this error is connected.
-90	Camera system error	This error is reported when a motion associated with the camera system has failed.
-91	Supply off	The external machine power supply is turned off and the digital input associated with the **SUPPLY_ON_IP** system configuration parameter is **false**. Alternatively, if the **SUPPLY_ON_ANY_IP** system configuration parameter is used, then the system cannot detect a logic level "1" on any of the digital inputs.
92	Emergency stop	The digital input associated with the **ESTOP_CHAIN_OK_IP** system configuration parameter is **false** and this indicates that the external machine emergency stop chain is broken and that machine power is turned off.
-93	Machine fault	The digital input associated with the **NO_MACHINE_FAULT_IP** system configuration parameter is false and this indicates that there is an external machine fault and that machine power is turned off.
94	Machine off	The digital input associated with the **MACHINE_ON_IP** system configuration parameter is **false** and this indicates that the machine power is turned off.
-95	Application already running	An attempt has been made to open the motion control library when has previously been opened by an earlier application.
-96..-105	OEM messages system action id 101..110	An OEM configured system action defined with an id from 101..110 has occurred.
-106	Invalid axis id	An operation has been requested than involves an axis (identified by its id / identifier) and either the axis id is invalid or the requested operation is not permitted with the axis.

4 Configuration files

The servo control handler is responsible for controlling the axes in a system. This is achieved using a servo control loop that is serviced at a rate defined by the user application. The servo control handler is able to control a wide range of axis types subject to appropriate configuration. For flexibility and ease of use, a user is able to perform this configuration by entering values for parameters in ASCII text files.

There are two configuration files defined for the control system: a system configuration file (**SYSTEM.CFG**) and an axis configuration file (**AXIS.CFG**). Examples of these files are provided with the DEVA004 card, they can be copied onto the user's system and modified to suit their requirements. Note that the demonstration programs provided with the card require these files to be located in specific directories. Users developing their own applications can specify different directories and filenames for their configuration files. Comments are permitted using either the ';' or '#' characters.

4.1 IO system definitions

Certain parameters in the system and axis configuration files define operations relating to physical inputs and outputs. For example, it is possible to configure axis overtravel limit switch inputs on any of the Deva004 opto-isolated or TTL digital inputs or on a 3rd party IO board. The system caters for devices connected to a variety of different IO systems and this section illustrates how the required functionality may be configured.

4.1.1 IO system types

The following table lists the available IO system types. Please contact Deva for details of other IO systems.

	Type	Description
0	USER_DEFINED_DIGITAL_IO	User defined digital IO system. The user writes the input status for the application software directly.
1	DEVA004_DIGITAL_IO	DEVA004 digital IO option. A 4 axis Issue 3.x PCI DEVA004, offers 16 inputs and 16 outputs (opto-isolated) for each installed card. If configured by **DIGITAL_IO_BOARD_TYPE** in the system configuration file, a 3rd party digital I/O board may offer a range of input and output configurations.
2	DEVA002_DIGITAL_IO	DEVA002 digital IO subsystem. Provides 32 inputs and 16 outputs. Requires installation of DEVA002 support software. This option is provided for legacy purposes only.
3	DEVA004_ANALOG_OUTPUT	DEVA004 analog output. Allows access to unused analog outputs. For example one channel is spare when using 3 servo axes with a 4-channel card.
4	DEVA004_AXIS_DIGITAL_IO	DEVA004 axis channel digital inputs. An Issue 3.x PCI card offers the following inputs: 0 Quadrature input A 1 Quadrature input B 2 Marker input M 3 Zero input Z 4 Limit input 0 5 Limit input 1
5	DEVA004_BOARD_EVENT_INPUT	DEVA004 board digital event inputs. An Issue 3.x PCI card provides a dedicated digital input event on the feature connector.
6	DEVA004_AXIS_EVENT_INPUT	DEVA004 axis digital event inputs. An Issue 3.x PCI card provides a dedicated digital input event for each axis on the feature connector.
7	DEVA004_TTL_IO	DEVA004 TTL inputs and outputs. An Issue 3.x PCI card provides TTL inputs and outputs on the feature connector.
8	DEVA004_PICS_IO	DEVA004 Renishaw PICS system inputs and outputs. An Issue 3.x PCI card provides signals on the PICS B and PICS A connectors.
9	DEVA004_DIGITAL_INPUT	An Issue 3.x PCI DEVA004, offers 16 inputs (opto-isolated) for each installed card.
10	DEVA004_DIGITAL_OUTPUT	An Issue 3.x PCI DEVA004, offers 16 outputs (opto-isolated) for each installed card.

4.1.2 IO configuration

To configure a physical input or output in the axis or system configuration file, a parameter is defined in the form:

PARAM = *system*, *number*, *input_logic*, [*io_module*]

The following table describes the elements of the configuration parameter:

Element	**Description**	
system	The type of IO system defined by **IO_SYSTEM_TYPES**	
	USER_DEFINED_DIGITAL_IO	PARAM = 0, *number*, *input_logic*
	DEVA004_DIGITAL_IO	PARAM = 1, *number*, *input_logic, superceded by* DEVA004_DIGITAL_INPUT & DEVA004_DIGITAL_OUTPUT
	DEVA002_DIGITAL_IO	PARAM = 2, *number*, *input_logic*
	DEVA004_ANALOG_ OUTPUT	PARAM = 3, *number*, *input_logic*
	DEVA004_AXIS_DIGITAL_IO	PARAM = 4, *number*, *input_logic*, [*io_module*]
	DEVA004_BOARD_EVENT_INPUT	PARAM = 5, *number*, *input_logic*, [*io_module*]
	DEVA004_AXIS_EVENT_INPUT	PARAM = 6, *number*, *input_logic*, [*io_module*]
	DEVA004_TTL_IO	PARAM = 7, *number*, *input_logic*
	DEVA004_PICS_IO	PARAM = 8, *number*, *input_logic*
	DEVA004_DIGITAL_INPUT	PARAM = 9, *number*, *input_logic*
	DEVA004_DIGITAL_OUTPUT	PARAM = 10, *number*, *input_logic*
number	0 = 1st input 1 = 2nd input 'n' = 'n'th input	
input_logic	0 = input signal logic not inverted 1 = input signal logic inverted	
io_module	Only used if **system** does not stack all the IO from 0 to 'n' or the **module** is not defined by the context in which the parameter is being used. For example, in the case of **DEVA004_BOARD_EVENT_INPUT**, this parameter identifies the 0 based index of the board for the specified **num**. However, if **DEVA004_AXIS_DIGITAL_IO** is defined in the context of the axis configuration file, the **io_module** parameter is automatically configured to the axis channel.	

For example, axis limit switches could be defined in the sub-section for a particular axis in the axis configuration file as follows:

```
POSOVERTRAVELIP = 1,5,0
NEGOVERTRAVELIP = 1,60
DATUMSWITCHIP = 7,2,1
```

This configuration would describe axis overtravel limit switches connected to input numbers 5 and 6 of the opto-isolated DEVA004 digital I/O system and a datum switch with inverted logic on input number 2 of the DEVA004 TTL input system on the feature connector.

The optional [m] parameter would be used to identify, for example, the channel or board specified by the **DEVA004_BOARD_EVENT_INPUT** or **DEVA004_AXIS_EVENT_INPUT**.

4.2 System configuration file

Basic operation of the servo control handler may be configured by setting the values of parameters in the system configuration file. Each record in the file consists of a pre-defined parameter name and a value for that parameter. Functions are provided in the library that allows the configuration file to be loaded when initialising the motion control system. The parameter values for the system configuration are written to an internal data structure of type **SYSTEM_CONFIG**. The values of this structure can be read and modified by the user's application using further functions provided in the library.

The following system configuration parameters are defined:

4.2.1 SIMULATION

If set to a value of 1, the system operates in simulation mode for development and demonstration purposes. Application software is able to run whether or not a card is installed and all axis motions are simulated. Simulation is disabled by setting the parameter to a value of 0.

4.2.2 UPDATE

To accurately control axis motion the servo control loop and model control functions performed by the servo control handler must be executed at regular intervals. The **UDPATE** parameter defines the interval between successive servo updates in units of (ns). Values in the range 1ms to 5ms are typical depending on the dynamic performance of the system axes and performance of the controlling PC.

4.2.3 AUTOINT

The on-board interrupt generator that is used to provide the required servo update interval may be enabled or disabled by setting the value of **AUTOINT** to 1 or 0

respectively. In normal use this parameter is set to 1 to enable on-board interrupts. If interrupts are disabled, the servo controller must be called on a regular basis via a user defined system.

4.2.4 PRBTMR

For the CMM motion controller this parameter defines the time (ms) for which the probe signal must be stable before a valid probe event is generated. The probe must be seated for 2x this period before it is re-enabled following a probe event. Permitted values range from 0 ms to 50 ms.

4.2.5 WDOGTMR

The motion control cards feature a watchdog timer that disables the axis drives and writes 0V to the analog outputs in the event of the system being shut down or crashing. This parameter defines the time (ms) after the system shuts down after which the watchdog trips. Permitted values range from 1 ms to the default value of 63 ms.

4.2.6 APPWATCHDOGTMR

The application watchdog timer may be used if it is required that the motion controller monitors execution of 3rd party software. This monitoring is only operational if the application watchdog timer is first enabled by the 3rd party software. If that software then hangs or fails to 'kick' the watchdog within the time (ms) defined by this parameter, then the system enters an emergency stop condition, motions in progress are aborted, the analog voltages to the drives are set to 0V and the drives disabled

4.2.7 DRIVEDISABLETMR

In an emergency condition, the axis drives are disabled and 0V is written to the analog outputs. In some cases, for example when disabling the axis drive causes an axis to coast, the drive disable timer can be used to delay the drive disable for a defined time (ms) after the analog output is set to 0V. This parameter can therefore ensure that the axis stops quickly and does not coast to a halt when set to an appropriate value.

4.2.8 DIGITAL_IO_BOARD_TYPE and DIGITAL_IO_BASE

These parameters configure a 3rd party digital I/O board which the motion control software will use to read inputs and write outputs. This allows users to connect dedicated signals such as **ESTOPIP, POSOVERTRAVELIP** and **NEGOVERTRAVELIP** to the digital I/O board. The following 3rd party boards are currently supported:

Parameter setting	Type	Description
DIGITAL_IO_BOARD_TYPE = 1	ADVANTECH_PCL730	Defines the 16 opto-isolated digital inputs and 16 opto-isolated digital outputs, PCL730 from Advantech.
DIGITAL_IO_BOARD_TYPE = 2	INTEL_8255	Defines control of a 3rd party IO board based on the Intel 8255. Provides a total of 16 inputs on ports A and C and 8 outputs on port B.

The base address of an ISA digital I/O board is defined in the **DIGITAL_IO_BASE** parameter. Please contact Deva for other supported I/O board types.

A digital I/O board defined with these parameters will correspond to the **DEVA004_DIGITAL_IO** I/O system type when configuring inputs.

4.2.9 PROBE_OVERSHOOT_DISTANCE

For the CMM motion controller this parameter is used in combination with joystick protection operation. If the probe is tripped and the axes can stop within this distance, then the axes will automatically 'bounce back' and retract along the approach vector until the probe is re-seated. If the axes stop further than this distance from the probe point, the automatic bounce back feature is disabled and the user must reverse the joystick direction to reseat the probe.

4.2.10 PROBE_CLEARANCE_DISTANCE

For the CMM motion controller this parameter is used in combination with joystick protection operation. If the probe tripped and then re-seated during the automatic bounce back, joystick control is resumed once the axes have retracted further than this distance from the probe point.

4.2.11 PROBE_JOYSTICK_PROTECT_DISTANCE

For the CMM motion controller this parameter is used in combination with joystick protection operation. This parameter defines the maximum distance that any axis may be moved from the probe point if the probe is not seated.

4.2.12 PROBE_JOYSTICK_PROTECT_FEED

For the CMM motion controller this parameter is used in combination with joystick protection operation. This parameter defines the maximum speed that may be applied to an axis through joystick control if the probe is not seated.

4.2.13 ESTOPIP

This parameter is now superseded by the **ESTOP_CHAIN_OK_IP and MACHINE_ON_IP** parameters that forms part of the machine power status system described in the following sections.

The parameter defines an input used to signal to the motion control software that the system is in an emergency stop condition. If the signal is at logic level 0 motions in progress are aborted, the analog voltages to the drives are set to 0V and the drives disabled.

See the section on **IO system definitions** for details on programming syntax for this parameter. The following example configures digital input 15 of the **DEVA004_DIGITAL_IO** system as the emergency stop input.

ESTOPIP = 1,15,0

4.2.14 SUPPLY_ON_IP & SUPPLY_ON_ANY_IP

These parameters allow the integrator to configure the system so that the motion controller can detect when the external machine control system power supply is turned on.

The **SUPPLY_ON_IP** parameter allows the user to specify a dedicated input signal for this purpose. The following example would be used in the case where digital input 10 of the **DEVA004_DIGITAL_IO** system is at logic level 1 when the external power supply is turned on:

SUPPLY_ON_IP = 1,10,0

Please see the section on **IO system definitions** for details on programming syntax for this parameter

If appropriate, an alternative method is to set the **SUPPLY_ON_ANY_IP** parameter to a non-zero value i.e.:

SUPPLY_ON_ANY_IP = 1

This means that the motion controller will determine that the external machine power supply is turned on if any of the **ESTOP_CHAIN_OK_IP**, **NO_MACHINE_FAULT_IP** and **MACHINE_ON_IP** inputs are at logic level 1.

4.2.15 ESTOP_CHAIN_OK_IP & ESTOP_CHAIN_OK_ENABLE_IP

In many typical machine control systems, a number of devices such as emergency stop buttons and the Deva004 hardware watchdog are connected in series to form an "emergency stop chain". Only when all of the devices in the chain are in the safe condition is the external machine control system able to apply power to the machine axis drives.

This parameter allows the integrator to configure an input that is connected to the end of the emergency stop chain. The motion controller can therefore detect when chain is broken and power removed from the machine axis drives The following example would be used in the case where digital input 12 of the **DEVA004_DIGITAL_IO** system is at logic level 1 when all of the devices in the chain are in the safe condition:

ESTOP_CHAIN_OK_IP=1,12,0

For special applications, it is possible to gate the checking of the emergency stop chain via the **ESTOP_CHAIN_OK_ENABLE_IP** digital input/output signal. In this case, the **ESTOP_CHAIN_OK_IP** signal is checked only when the **ESTOP_CHAIN_OK_ENABLE_IP** is set.

Please see the section on **IO system definitions** for details on programming syntax for these parameters

4.2.16 NO_MACHINE_FAULT_IP

This parameter allows the integrator to configure an input that indicates power is not applied to the machine axis drives due to a fault condition in the external machine control system. Typically, some remedial action is required to reset the fault before the operation can continue. The following example would be used in the case where digital input 14 of the **DEVA004_DIGITAL_IO** system is at logic level 1 when there is no fault:

NO_MACHINE_FAULT_IP=1,14,0

Please see the section on **IO system definitions** for details on programming syntax for this parameter

4.2.17 MACHINE_ON_IP

This parameter allows the integrator to configure an input that indicates the external machine control system has applied power to the machine axis drives and the machine is turned of. The following example would be used in the case where digital input 15 of the **DEVA004_DIGITAL_IO** system is at logic level 1 when the machine is turned on:

MACHINE_ON_IP=1,15,0

Please see the section on **IO system definitions** for details on programming syntax for this parameter

4.2.18 SYSTEM_ACTION

This parameter may be used one or more times to configure a range of system actions that determine operation of the motion control system. For example, it might be required to configure a particular digital input to cause an emergency stop or to configure an output that is enabled when an axis drive is enabled. The

SYSTEM_ACTION is configured by defining a source parameter and destination parameter.

A source can be a digital input, control system flag such as feedhold or hardware event such as a probe. If the source condition is true, the destination parameter is activated. The destination can be a digital output, control system flag such as emergency stop or hardware event output.

A **SYSTEM_ACTION** definition requires 10 parameters, 5 each for the source 's' and destination 'd':

SYSTEM_ACTION = id,s0,s1,s2,s3,s4,d0,d1,d2,d3,d4

4.2.18.1 System action id

The **id** is a non-zero number that may be used by the application software to identify the particular action.

4.2.18.2 System action source / destination parameters

The parameters themselves are defined in the following manner for both source and destination:

s/d0	s/d1	s/d2	s/d3	s/d4
1 DIGITAL_IO_ACTION	0 USER_DEFINED_DIGITAL_IO 1 DEVA004_DIGITAL_IO 2 DEVA002_DIGITAL_IO 3 DEVA004_ANALOG_OUTPUT	Number	1 Invert	
	4 DEVA004_AXIS_DIGITAL_IO	Number	1 Invert	Channel
	5 DEVA004_BOARD_EVENT_INPUT	Number	1 Invert	Board
	6 DEVA004_AXIS_EVENT_INPUT	Number	1 Invert	Channel
	7 DEVA004_TTL_IO	Number	1 Invert	
	8 DEVA004_PICS_IO	Number	1 Invert	Board
	9 DEVA004_DIGITAL_INPUT	Number	1 Invert	
	10 DEVA004_DIGITAL_OUTPUT	Number	1 Invert	
2 EVENT_ACTION	1 MARKER_EVENT_TYPE 2 ZERO_EVENT_TYPE	Channel		
	3 PROBE_EVENT_TYPE	Board		
	4 TIMER_EVENT_TYPE	Timer		
	5 PULSE_GENERATOR_EVENT_TYPE 6 AXIS_COMPARE_EVENT_TYPE	Channel		
	7 BOARD_INPUT_EVENT_TYPE	Board		
	8 AXIS_INPUT_EVENT_TYPE	Channel		
	9 USER_REQUEST_EVENT_TYPE	Board		
3 CONTROL_SYSTEM_ACTION	1 EMERGENCY_STOP_SYSTEM_TYPE[1]	Watchdog		
	2 MOTION_ABORT_SYSTEM_TYPE	Handle to motion		
	3 FEEDHOLD_SYSTEM_TYPE 4 ANY_DRIVES_ENABLE_SYSTEM_TYPE 5 ALL_DRIVES_ENABLE_SYSTEM_TYPE 6 AXIS_EMERGENCY_SYSTEM_TYPE			
	7 POWER_STATUS_TYPES[2]	Type		
	8 APPLICATION_RUNNING_SYSTEM_TYPE			
4 LOGIC_SYSTEM_ACTION	0 FALSE_LOGIC_SYSTEM_TYPE			
	1 TRUE_LOGIC_SYSTEM_TYPE			

Note[1] If the system action **EMERGENCY_STOP_SYSTEM_TYPE** is used as a destination parameter, by default the Deva004 watchdog contacts open momentarily when the system action is triggered. If required, setting the d2 parameter to 1 will prevent the watchdog contacts from opening.

If the id of the **EMERGENCY_STOP_SYSTEM_TYPE** system action used as a destination parameter is in the range 1..10, then the error message "Emergency stop system action id 1..10" will be displayed when the system action is triggered. If the id is in the range 101.,110 then the error message "OEM message id 101..110" will be displayed. These error messages are stored in the "message.txt" file normally located in the "windows" folder and may be modified by the system integrator as required.

Note[2] If the system action selects **POWER_STATUS_TYPES**, then the s2/d2 parameter represents which of the available power status types is involved in the system action:

SUPPLY_OFF_POWER_STATUS_TYPE = 0

EMERGENCY_STOP_POWER_STATUS_TYPE = 1

MACHINE_FAULT_POWER_STATUS_TYPE = 2

MACHINE_OFF_POWER_STATUS_TYPE = 3

MACHINE_ON_POWER_STATUS_TYPE = 4

4.2.18.3 System action example

As describe above, a **SYSTEM_ACTION** definition requires 10 parameters, 5 each for the source 's' and destination 'd':

SYSTEM_ACTION = id,s0,s1,s2,s3,s4,d0,d1,d2,d3,d4

For example, the following entry could be used to define a SYSTEM_ACTION with and id of 9 that forces an emergency stop when DEVA004_DIGITAL_IO input number 10 is high:

SYSTEM_ACTION = 9, 1, 1, 10, 0, 0, 3, 1, 0, 0, 0

Using an id of 9 means that the error message "Emergency stop system action id 9" will be displayed.

4.2.19 POSN_CALC_TOLERANCE

This parameter allows the user to define the position calculation tolerance used by the motion controller. This parameter is optional and is only required in unusual circumstances when the default value (in the order of $1x10^{-9}$ system units) is not satisfactory

The value of this parameter and values derived from it are used to define the limit at which values of other parameters are considered equal. As such, it accounts for the small calculation inaccuracies inherent in the use of floating point mathematics by the motion controller.

Typical motion control axes might have a feedback resolution ranging from 0.0001 to 0.1 mm or inches determined by the **PITCH** and **COUNT** parameters of the axis configuration file. If values of these parameters were adjusted so that the axes resolutions were much larger, it might be necessary to set a proportionally larger value of **POSN_CALC_TOLERANCE**.

4.3 Axis configuration file

The axes to be controlled by the servo control handler may be configured by setting the values of parameters in the axis configuration file. Each record in the file consists of a pre-defined parameter name and a value for that parameter.

Functions are provided in the library to allow the configuration file to be loaded when initialising the motion control system. The configuration parameter values for each axis are written to an internal data structure of type **REAL_AXIS_CONFIG**. The values of this structure can be read and modified by the user's application, and the configuration file can be updated with new parameter values. Again, functions are provided in the library to facilitate this.

Configuration data for each axis is headed by a record that sets the axis name i.e. 'NAME=x'. Proceeding records set values to parameters for the named axis. Additional axes may be appended in the same manner.

The following axis configuration parameters are defined:

4.3.1 NAME

The parameter **NAME** defines the name of the axis that is characterised by proceeding records. The axis name is a single ASCII character that may be used in the user's application if required.

4.3.2 TYPE

Parameter **TYPE** defines the type of the axis.

Parameter setting	Type	Description
TYPE = 0		Default
TYPE = 1	DIAMETER_AXIS_FLAG	Not implemented.
TYPE = 2	BACKLASH_TAKEUP_AXIS	Identifies that the axis is to be used primarily to take out the backlash on another axis at the start of a motion.
TYPE = 4	MEASUREMENT_AXIS	The axis is not controlled and is for measurement use only.
TYPE = 8	VIRTUAL_AXIS_FLAG	The axis is a virtual, software only axis without any physical connections.
TYPE = 16	JOYSTICK_CONTROLLED_AXIS_FLAG	The axis is controlled by an operator using a joystick. Although full control is not possible, programmed moves using combinations of 2 or less axes are supported.

4.3.3 DRIVE_ENABLE_FLAGS

Parameter **DRIVE_ENABLE_FLAGS** optionally configures operation of the axis drive enable output. To use this parameter, a bit-pattern is constructed where a logical '1' in a particular position enables a function.

Parameter setting	Type	Description
DRIVE_ENABLE_FLAGS = 1	DRIVE_ENABLE_INVERT_TYPE	Inverts the sense of the axis drive enable output so that it is set to logic level 0 when the user application enables the axis.

The default value for this parameter is DRIVE_ENABLE_FLAGS = 0 which specifies no inversion.

4.3.4 FEEDBACK_MONITOR_FLAGS

Parameter **FEEDBACK_MONITOR_FLAGS** defines the monitoring that is applied to the axis feedback for detection of disconnection errors, quadrature failure, power failure and so on. To use this parameter, a bit-pattern is constructed where a logical '1' in a particular position enables a particular feedback monitor function.

Parameter setting	Type	Description
FEEDBACK_MONITOR_FLAGS = 1	FEEDBACK_DISCONNECT_MONITOR_TYPE	Monitors the axis channel for disconnection by checking the state of the A, B and M input signals.
FEEDBACK_MONITOR_FLAGS = 2	FEEDBACK_QUADRATURE_ERROR_MONITOR_TYPE	Detects when the A and B quadrature signals have jumped to an illegal state.
FEEDBACK_MONITOR_FLAGS = 4	FEEDBACK_PARITY_ERROR_MONITOR_TYPE	Detects invalid data on an SSI encoder interface channel.

For example, setting FEEDBACK_MONITOR_FLAGS = 3 would enable monitoring of both disconnection and quadrature failure. The default value for this parameter is FEEDBACK_MONITOR_FLAGS = 1 which specifies monitor quadrature failure only.

4.3.5 CHANNEL

Each controlled axis in a system requires an encoder input and analog output. The parameter **CHANNEL** is used to configure the hardware channel allocated by a user to the axis. Values in the range 0 to 15 may be defined with up to 16 axes on four boards. Correspondence with the connectors defined in the hardware manual is as follows:

Channel	Board	Encoder input	Analog output	Servo enable
0	1	X	Anx	Enx
1	1	Y	Any	Eny
2	1	Z	Anz	Enz
3	1	S	Ans	Ens
4	2	X	Anx	Enx
5	2	Y	Any	Eny
6	2	Z	Anz	Enz
7	2	S	Ans	Ens
8	3	X	Anx	Enx
9	3	Y	Any	Eny
10	3	Z	Anz	Enz
11	3	S	Ans	Ens
12	4	X	Anx	Enx
13	4	Y	Any	Eny
14	4	Z	Anz	Enz
15	4	S	Ans	Ens

4.3.6 PITCH and COUNT

The **PITCH** and **COUNT** parameters are used by the servo control handler to scale encoder counts to real world units.

The parameter **COUNT** defines the number of quadrature encoder counts produced by an axis travelling the distance defined by the **PITCH** parameter. The user should select values that provide the correct encoder scaling and ensure that consistent units are employed.

For example, consider an axis with a 2000 line encoder on the end of a 0.2in pitch ball screw. This axis would produce 8000 quadrature counts for every 0.2in of travel produced by one pitch of the screw. An axis with a 0.01mm linear scale would produce 100 quadrature counts for every 1mm pitch travelled by the reader head of the scale.

Changing the sign of the **PITCH** parameter will invert the direction of axis count. Before closing the control loop, please ensure that the analog voltage to the axis drive and the axis count direction are phased correctly. See the description of the **MAXVOLT** parameter for more information.

When using a stepper control card, in a similar way the value of **PITCH** represents how far the axis moves for every **COUNT** steps of the stepper system.

4.3.7 AUXCHANNEL

The parameter permits definition of a second, auxiliary encoder channel for the axis. The auxiliary encoder channel resolution is defined by the **AUXPITCH** and **AUXCOUNT** parameters.

Please refer to the section on **CHANNEL** for information on channel numbering.

4.3.8 AUXPITCH and AUXCOUNT

The **AUXPITCH** and **AUXCOUNT** parameters are used by the servo control handler to scale a second, auxiliary encoder defined by **AUXCHANNEL** to real world units.

There is a direct correspondence between these parameters and the **PITCH** and **COUNT** parameters of the primary encoder channel. Please refer to the section on **PITCH** and **COUNT** for information on setting the value of these parameters.

4.3.9 ANALOGCHANNEL

If two or more configured axes share one axis drive or an axis is controlled by an analog output other than the one selected by the **CHANNEL** parameter, this parameter is used to indicate that the axis will be controlled using the analog output of the specified **ANALOGCHANNEL**. This feature could be used if one motor was switched between two or more physical axes by a system of clutches.

If a stepper motor controller is installed, this parameter selects the stepper motor axis channel. Please refer to the section on **ANALOGCHANNEL_PITCH** and **ANALOGCHANNEL_COUNT** for information on how to configure the stepper axis step size.

Please refer to the section on **CHANNEL** for information on channel numbering.

4.3.10 ANALOGCHANNEL_PITCH and ANALOGCHANNEL_COUNT

If the **ANALOGCHANNEL** parameter is used to select a stepper motor axis channel, the **ANALOGCHANNEL_PITCH** and **ANALOGCHANNEL_COUNT** parameters may be used by the servo control handler to scale stepper motor steps to real world units.

If these parameters are not defined, then the system uses the values defined by the **PITCH** and **COUNT** parameters to scale the stepper system steps. Otherwise, the value of **ANALOGCHANNEL_PITCH** represents how far the axis moves for every **ANALOGCHANNEL_COUNT** steps of the stepper system.

4.3.11 MAXVOLT

The maximum analog voltage applied to the axis drive is scaled by the parameter **MAXVOLT** (mV). It is recommended that the user adjust the axis drive system so that the maximum analog voltage applied to the axis drive is between 6000mV and 9000mV.

If the servo controller performs the position loop and the axis drive amplifier the speed loop, **MAXVOLT** is the voltage that causes the axis to move at the speed defined by the **MAXSPEED** parameter. As an alternative to **MAXSPEED** the parameter **DRIVESCALE** can define the drive speed scaling value. This is useful if the maximum speed of the axis that can be requested by application software is substantially lower than can be achieved by the axis.

If the servo controller performs the position loop and the speed loop is enabled with a non-zero **VP**, **MAXVOLT** is the voltage that causes the axis to accelerate at the rate defined by the **MAXACCEL** parameter. As an alternative to **MAXACCEL** the parameter **DRIVESCALE** can define the drive acceleration scaling value. This is useful if the maximum acceleration of the axis that can be requested by application software is substantially lower than can be achieved by the axis.

If a positive analog voltage to the axis drive causes the axis to move and count in a positive direction, the sign of the **MAXVOLT** parameter should be positive. The **MAXVOLT** parameter may be set to a negative value if a negative voltage to the axis drive moves the axis in a positive direction. Before closing the control loop, please ensure that the analog voltage to the axis drive and the axis count direction are phased correctly. See the description of the **PITCH** parameter for more information.

4.3.12 DRIVESCALE

If the servo controller performs only the position loop and the axis drive amplifier the speed loop, **DRIVESCALE** defines the speed that the axis moves when an analog voltage defined by parameter **MAXVOLT** is applied to the axis drive. This parameter is provided as an alternative to the **MAXSPEED** parameter**.**

If the servo controller performs the position loop and the speed loop is enabled with a non-zero **VP**, **DRIVESCALE** defines the acceleration that the axis achieves when an

analog voltage defined by parameter **MAXVOLT** is applied to the axis drive. This parameter is provided as an alternative to the **MAXACCEL** parameter**.**

4.3.13 DRIVEOFFSET

The **DRIVEOFFSET** parameter (mV) can be used to offset the analog output voltage applied to the analog reference of the axis drive amplifier. The purpose of the offset is to counteract any inherent offsets in the axis drive amplifier that cause the axis to move or 'drift' when the servo controller requires the axes to be stationary.

4.3.14 DBMVOLT

Certain types of axis drive amplifiers have a deadband and do not respond to analog voltages below a threshold voltage. The analog voltage applied to the axis drive that moves the axis at the slowest, detectable speed is defined by the deadband voltage parameter **DBMVOLT** (mV). It is assumed that the deadband, if any, is symmetrical and the value of this parameter is applied in both positive and negative directions. The sign of the **DBMVOLT** parameter should therefore be positive.

4.3.15 MAXSPEED

The parameter **MAXSPEED** defines the maximum permitted speed of the axis (units/s) that may be requested by application software.

If the **DRIVESCALE** parameter is set to zero or not defined and the servo controller performs only the position loop and the axis drive amplifier the speed loop, **MAXSPEED** also defines the speed that the axis moves when an analog voltage defined by parameter **MAXVOLT** is applied to the axis drive.

4.3.16 PREDATUM_MAXSPEED

The parameter **MAXSPEED** defines the maximum permitted speed of the axis (units/s) before the axis datum cycle has been completed. This parameter allows the speed of the axis to be restricted until software over-travel protection has been activated by completion of the datum cycle.

4.3.17 MAXACCEL

The servo controller employs either a linear or 'S' curve acceleration and deceleration ramp during axis motions. The **MAXACCEL** parameter defines the maximum acceleration of the axis (unit/s/s) that may be requested by application software. If the **MAXDECEL** parameter is zero or not defined, the value defined by **MAXACCEL** also defines the maximum deceleration.

If the **DRIVESCALE** parameter is set to zero or not defined, the servo controller performs the position loop and the speed loop is enabled with a non-zero **VP**,

MAXACCEL also defines the rate at which the axis accelerates when an analog voltage defined by parameter **MAXVOLT** is applied to the axis drive.

4.3.18 MAXDECEL

The servo controller employs either a linear or 'S' curve acceleration and deceleration ramp during axis motions. The **MAXDECEL** parameter defines the maximum deceleration of the axis (unit/s/s) that may be requested by application software. If the **MAXDECEL** parameter is zero or not defined, the value defined by **MAXACCEL** also defines the maximum deceleration.

4.3.19 MAXJERK

The **MAXJERK** parameter defines the maximum jerk or rate of change of acceleration of the axis (units/s/s/s) that may be applied by the control system. Use of this parameter will result in 'S' curve velocity profiles and may be disabled by setting a value of 0.0.

4.3.20 EMERGACCEL, EMERGDECEL and EMERGJERK

In certain emergency stop situations, for example when a probe collision is detected on a CMM machine, the control system may optionally use an alternative set of parameters **EMERGACCEL**, **EMERGDECEL**, **EMERGJERK**. Each of these alternate parameters will override the corresponding **MAXACCEL**, **MAXDECEL** or **MAXJERK** values if defined in the axis configuration file.

The values should be chosen according to the requirements and performance of the mechanical system.

4.3.21 KV

The proportional gain applied to the axis control loop is defined by the parameter **KV** (/s). The value of parameter **KV** is empirical and is generally optimised by the user in order to produce the required axis response. Values of **KV** in the range 10/s to 200/s are typical.

Too high a value of **KV** will produce instability and oscillation whilst too low a value will produce a sluggish response. For a correctly characterised axis, the following error is directly related to speed by **KV**:

following error (units) = speed (units/s) / KV (/s)

The analog reference to the axis drive amplifier should be adjusted to provide the correct following error. For example, a following error of 1mm would indicate a correctly adjusted axis travelling at 10mm/s with a KV of 10/s.

4.3.22 KD

The **KD** parameter sets the derivative gain of the control system. This term improves the dynamic response by providing a damping effect in systems where high inertia coupled with low damping (low friction) can lead to instability. The term is dimensionless and should be applied cautiously. Fractional values in the range 0.1 through 2.0 have been found to provide a significant improvement to axis performance.

A value of 0.0 may be used to disable this feature.

4.3.23 KI and LIMITKI

The **KI** parameter sets the integral gain of the control system. The term improves the steady state performance of the control system in the presence of disturbances such as friction, offset drift, and gravity. The term has units of /s/s and can lead to instability if used excessively. The **LIMITKI** parameter is used to limit the effect of the **KI** term and is defined as a percentage of maximum output speed. This prevents the integral term building up and delivering full speed demand under axis stall conditions.

A value of 0.0 may be used to disable this feature.

4.3.24 VP

The **VP** (/s) parameter sets the value of speed loop proportional gain for axes with drive amplifiers that operate in torque or current mode. A non-zero value enables the speed loop that determines the value of analog voltage applied to the driver amplifier as a product of **VP** and the velocity error.

Too high a value of **VP** will produce instability and oscillation whilst too low a value will produce a sluggish response. It is to be expected that, for stability of a correctly scaled system, the position loop gain **KV** is lower than the speed loop gain **VP**.

If **AUXCHANNEL**, **AUXPITCH** and **AUXCOUNT** are configured the system will operate with the actual velocity determined from the auxiliary encoder channel. This dual loop facility may be used if there are separate position and velocity feedback devices such as a linear scale mounted on the machine and an encoder integral with the motor.

A value of 0.0 may be used to disable the speed loop control where the speed loop is provided by the axis drive.

4.3.25 VD

If a non-zero **VP** enables the speed loop, the **VD** parameter sets the value of speed loop derivative gain for axes with drive amplifiers that operate in torque or current mode. This term improves the dynamic response by providing a damping effect in systems where high inertia coupled with low damping (low friction) can lead to instability. The term is dimensionless and should be applied cautiously.

A value of 0.0 may be used to disable this feature.

4.3.26 VI and LIMITVI

If a non-zero **VP** enables the speed loop, the **VI** parameter sets the value of speed loop integral gain. The term improves the steady state performance of the speed loop in the presence of disturbances such as friction, offset drift, and gravity. The term has units of /s/s and can lead to instability if used excessively. The **LIMITVI** parameter is used to limit the effect of the **VI** term and is defined as a percentage of maximum output acceleration. This prevents the integral term building up and delivering full speed demand under axis stall conditions.

A value of 0.0 may be used to disable this feature.

4.3.27 KS

This parameter is now obsolete; please refer to **VP, VD** and **VI** for details of how to implement a speed loop in the servo controller.

The **KS** (Vs/mm) parameter sets the value of speed loop gain for axes with drive amplifiers that operate in torque mode. A non-zero value enables the speed loop that determines the value of analog voltage applied to the driver amplifier as a product of **KS** and the velocity error. In simplified form, the control loop may be expressed as:

Analog (V) = (following error (units) x KV (/s) – actual velocity (units/s)) x KS (Vs/mm)

It can be seen that the **MAXVOLT** parameter has no effect when the speed loop is in use and that it is therefore necessary to change the sign of **KS** if the phasing of the drive system is inverted.

If **AUXCHANNEL**, **AUXPITCH** and **AUXCOUNT** are configured the system will operate with the actual velocity determined from the auxiliary encoder channel. This dual loop facility may be used if there are separate position and velocity feedback devices such as a linear scale mounted on the machine and an encoder integral with the motor.

A value of 0.0 may be used to disable this feature.

4.3.28 VFF

The **VFF** parameter is used to set the Velocity Feed Forward term in the control system. The parameter defines a contribution to the output that is generated from demanded speed without reference to the feedback system (open loop). Increasing this parameter from 0.0 has the effect of reducing the following error or lag of the system. However, as **VFF** is increased, greater reliance is placed on the accuracy and stability of servo amplifier parameters such as the voltage/speed constant. Allowed values of **VFF** are from 0.0 to a maximum value 1.0 with typical values in the range 0.25 to 0.75.

A value of 0.0 may be used to disable this feature.

4.3.29 AFF

If a non-zero **VP** enables the speed loop, the **AFF** parameter is used to set the Acceleration Feed Forward term in the speed loop. The parameter defines a contribution to the output that is generated from demanded acceleration without reference to the feedback system (open loop). Increasing this parameter from 0.0 has the effect of reducing the speed error or lag of the system during the acceleration phases of a motion. However, as **AFF** is increased, greater reliance is placed on the accuracy and stability of servo amplifier parameters such as the voltage/torque constant. Allowed values of **AFF** are from 0.0 to a maximum value 1.0 with typical values in the range 0.25 to 0.75.

A value of 0.0 may be used to disable this feature.

4.3.30 ACTIVATE_CONTROL_DEADBAND, MAINTAIN_CONTROL_DEADBAND and CONTROL_DEADBAND_FLAGS

The purpose of the control deadband is to disable axis control when the axis arrives within the band specified by the **ACTIVATE_CONTROL_DEADBAND**. Control remains disabled unless the axis moves outside of the **MAINTAIN_CONTROL_DEADBAND** or a move is started.

The **MAINTAIN_CONTROL_DEADBAND** can optionally be left undefined, set to a larger value than **ACTIVATE_CONTROL_DEADBAND** to provide some hysteresis or set to zero.

The concept is that **ACTIVATE_CONTROL_DEADBAND** is set to the position band that the machine can achieve and that **MAINTAIN_CONTROL_DEADBAND** may optionally be set to the position band that the application requires. For example, the machine may reliably achieve target position to within 2um but hunt within this region. So, **ACTIVATE_CONTROL_DEADBAND** would be set to 0.002. However, due to the requirements of the application it may be only necessary that the axis remains stationary within a 10um band around target. The **MAINTAIN_CONTROL_DEADBAND** would then be set to 0.010. If the axis drifted or was moved outside this band, the control would reactivate until the axis returned to within **ACTIVATE_CONTROL_DEADBAND**.

Optionally, the **CONTROL_DEADBAND_FLAGS** parameter may be used to change operation of the control deadband. To use this parameter, a bit-pattern is constructed where a logical '1' in a particular position enables a particular feature.

Parameter setting	Type	Description
CONTROL_DEADBAND_FLAGS = 1	CONTROL_DEADBAND_ZERO_SPEED_REFERENCE_TYPE	Sets the analog speed reference to zero when the axis is in the control deadband.
CONTROL_DEADBAND_FLAGS = 2	CONTROL_DEADBAND_DRIVE_DISABLE_TYPE	Disables the axis drive when the axis is in the control deadband.

For example, if CONTROL_DEADBAND_FLAGS = 3, both the analog speed reference is set to zero and the axis drive disabled when the control deadband is active. The default value for this parameter is CONTROL_DEADBAND_FLAGS = 1 which specifies zero analog speed reference only.

4.3.31 SRVERR and MINSRVERR

The parameters **SRVERR** and **MINSRVERR** are used by the servo control handler to monitor axis position during a motion and at standstill respectively.

A servo error will be generated during a motion if the actual position of an axis varies from its expected position by more than the distance specified in parameter **SRVERR** (units). Axis motion is therefore disabled if the system detects that the axis has run away or stalled.

The standstill servo error protection band is specified by parameter **MINSRVERR** (units). A servo error will be generated and the axis disabled if the axis moves outside this band.

The user should select values for these parameters that provides the required level of protection depending on the application and axis performance.

4.3.32 POSNBAND and POSNTIME

The parameters **POSNBAND** and **POSNTIME** are used by the servo control handler to detect whether or not the axis has reached its programmed target position.

The parameter **POSNBAND** (units) defines an 'in-position' band around the target position of an axis. The servo control handler considers an axis to be at the programmed target position when the actual axis position lies within this band.

After completion of an axis motion the servo control handler waits for the axis to reach its target position. A servo error is generated if this position is not reached within a time specified by parameter **POSNTIME** (s).

4.3.33 POSLIMIT and NEGLIMIT

The parameters **POSLIMIT** (units) and **NEGLIMIT** (units) define software overtravel limits in the positive and negative directions respectively.

If the user attempts to drive the axis beyond either of these limits, the servo control handler will decelerate the axis to a stop and a servo error will be generated. Software overtravel limits are only enabled following completion of an axis reference cycle or after the axis has been set to a position using the **preset_axis** function.

A value of 0.0 in both **POSLIMIT** and **NEGLIMIT** may be used to disable the protection.

4.3.34 DATUM and DATUMMODE

Parameter **DATUM** defines the axis position (units) that is loaded after the axis has completed a reference or datum cycle. Parameter **DATUMMODE** defines the behaviour of the axis when searching for the reference switch and marker during a reference or datum cycle.

Parameter setting	Type	Description
DATUMMODE = 0	NO_DATUM_CYCLE	No datum cycle is defined for the axis.
DATUMMODE = 1	DATUM_SWITCH_POSITIVE	The axis will accelerate to the speed defined by parameter **DATSWITCHSPEED** in the axis configuration file and move in a positive/negative direction until the normally closed reference switch defined by parameter **DATUMSWITCHIP** is open circuit. The axis will then reverse and move at the speed defined by parameter **DATMARKSPEED** until clear of the reference switch. The axis will continue to move at this speed until the next encoder marker pulse is encountered at which point the axis position is set to the value defined by parameter DATUM. The axis will then decelerate to a stop.
DATUMMODE = 2	DATUM_SWITCH_NEGATIVE	
DATUMMODE = 3	DATUM_NO_SWITCH_POSITIVE,	The axis will accelerate to the speed defined by parameter **DATMARKSPEED** in the axis configuration file and move in a positive/negative direction until the next encoder marker pulse is encountered at which point the axis position is set to the value defined by parameter **DATUM** in the axis configuration file. The axis will then decelerate to a stop.
DATUMMODE = 4	DATUM_NO_SWITCH_NEGATIVE	
DATUMMODE = 5	DATUM_MANUALLY	The axis datum is defined by the application software after manually positioning the axis.
DATUMMODE = 6	DATUM_SWITCH_POSITIVE_NO_MARKER,	The axis will accelerate to the speed defined by parameter **DATSWITCHSPEED** in the axis configuration file and move in a positive/negative direction until the normally closed reference switch defined by parameter **DATUMSWITCHIP** is open circuit. The axis will then reverse and move at the speed defined by parameter **DATMARKSPEED** until clear of the reference switch at which point the axis position is set to the value defined by parameter DATUM. The axis will then decelerate to a stop.
DATUMMODE = 7	DATUM_SWITCH_NEGATIVE_NO_MARKER	

4.3.35 DATUMSWITCHIP, DATSWITCHSPEED and DATMARKSPEED

Parameter **DATUMSWITCHIP** defines the system input used for the axis reference switch. See the section on **IO system definitions** for details on programming this parameter.

Parameter **DATSWITCHSPEED** defines the speed at which the axis moves while searching for the reference switch during a datum cycle.

Parameter **DATMARKSPEED** defines the speed at which the axis moves while searching for the reference marker during a datum cycle.

4.3.36 POSOVERTRAVELIP, NEGOVERTRAVELIP

Parameters **POSOVERTRAVELIP** and **NEGOVERTRAVELIP** define the inputs used for the positive and negative axis overtravel switches.

The axis will stop when an overtravel input is activated and the axis is moving in the appropriate direction. Further motion in the direction of the overtravel input is prohibited. The parameter **OVERTRAVELIPMODE** defines the manner by which the system brings the axis to a stop.

See the section on **IO system definitions** for details on programming these parameters.

4.3.37 OVERTRAVELIPMODE

The parameter **OVERTRAVELIPMODE** defines the way that the system responds to the inputs used for the positive and negative axis overtravel switches.

Parameter setting	Type	Description
OVERTRAVELIPMODE = 0	HARDWARE_OVERTRAVEL_STANDARD_DECEL_TYPE	The axis stops using the standard deceleration when the axis overtravel limit switch input is activated.
OVERTRAVELIPMODE = 1	HARDWARE_OVERTRAVEL_EMERGENCY_DECEL_TYPE	The axis stops using the emergency deceleration specified by the **EMERGDECEL** parameter when the axis overtravel limit switch input is activated.
OVERTRAVELIPMODE = 2	HARDWARE_OVERTRAVEL_AXIS_EMERGENCY_TYPE	An emergency is generated when the axis overtravel limit switch input is activated. The analog output to the axis drive is clamped to 0V and the axis drive disabled.

4.3.38 SKEW_PAIRED_CHANNEL, SKEW_GAIN and SKEW_LIMIT

These parameters are provided for control of machines that have a pair of motors driving one axis. This arrangement can often be found on gantry CMM and milling

machines. One motor and feedback device is mounted on each side of the gantry and the motors are controlled so that the gantry does not skew during motions.

Two axes must be defined in the axis configuration file and the control loop of each axis is directly affected by the position of the other axis. For each of the two axes **SKEW_PAIRED_CHANNEL** identifies the channel with which the axis is paired.

The **SKEW_GAIN** parameter defines how much effect one axis has on the other. For example, if one axis is ahead of the other, there is an increase in speed demand to the slower axis. At the same time the speed demand to the faster axis is reduced. Typical values of **SKEW_GAIN** range from 0.0 to 1.0.

If the axes are skewed by an amount bigger than defined by the **SKEW_LIMIT** (units) parameter, a servo error will generated and the axes disabled.

The following extract from a configuration file demonstrates the inter-relationship between skew axes:

```
NAME=x
CHANNEL=0
SKEW_PAIRED_CHANNEL=3
SKEW_GAIN=0.5
SKEW_LIMIT=2

NAME=u
CHANNEL=3
SKEW_PAIRED_CHANNEL=0
SKEW_GAIN=0.5
SKEW_LIMIT=2
```

The axis definitions are symmetrical so that moves may be commanded in either of the axes. The application software only needs to register and control one of the two axes, the servo controller will automatically enable and control the other axis. This ensures that application software does not need to change if a gantry axis is implemented.

In many gantry axes the motors run in opposite directions and the feedback counts are similarly inverted. This **MUST** be taken account of in the axis configuration file. Furthermore, it is expected that the speed and gain parameters for the two axes will be identical so that the motion characteristics will be similar.

AxisTuner.exe may be used as normal to tune the gantry axes independently. If mechanics permit, it may be helpful to perform initial configuration without skew control enabled and only one motor connected at a time. Also, it may be helpful to set the **SKEW_GAIN** to 0 so that the axis loops do not interact directly. The aim is that the following error for both axes is matched.

After power-up the axis is assumed to be in an un-skewed state and the application software may request motions or a datum cycle to be performed. The axis is not de-skewed until after the first datum cycle has been completed. The **DATUM** values in

the axis configuration file define the position of each of the skew axes at the marker pulse with the machine in an un-skewed condition.

It may be helpful to manually adjust the feedback devices so that the marker pulses are close together when the machine is not skewed.

4.3.39 SIMUL

Simulation of an axis may be enabled or disabled by setting the value of the **SIMUL** parameter to 1 or 0 respectively. If an axis is simulated, the axis analog voltage is clamped to 0V and encoder counts are simulated by the system. In this way it is possible to simulate axis motions without a physical axis connection. In normal operation, axis simulation is disabled and the physical axes are controlled.

5 PCI motion control hardware

5.1 Functional description

The DEVA004 is designed to control up to four servo motors via incremental encoders or linear scales feedback using a PCI based PC computer. The following sections describe the various functions of the interface. For a detailed description of how to program these facilities, please refer to section 3.

5.1.1 Quadrature input

Each encoder channel has three differential / single ended input circuits designated A, B and M. The A and B inputs accept the quadrature signals from the encoder and drive an up/down counter via a x4 directional discriminator circuit. The counter has 20 bit resolution and may be read at any time and is typically sign extended in software as required. The maximum count rate is in excess of 10 MHz.

5.1.2 Marker input

The M input circuit designated marker is a positive edge triggered input which can synchronously latch the counter value. To gain the most accurate result the input conditioning circuit latches the counter when both A and B inputs are at a logic one level. The user must therefore phase the A and B signals carefully to meet this criteria. Should this not be achievable circuit operation is still possible although the latched value will not be accurate to a single count. Correct phasing may also be achieved by using the input invertors on signals A, B & M.

To use the marker input the marker circuit must be enabled. Once a positive edge has occurred on the marker input the counter is latched when both A and B are at logic one. The marker latch register may now be read while the counter continues to maintain position. To enable the maker latch to capture a further marker event the marker function should be disabled and then re-enabled. Marker synchronisation with signals A & B can optionally be disabled.

5.1.3 Renishaw probe interface

The DEVA004 incorporates two Renishaw Product Interconnection System (PICS) compatible probe inputs.

Input PICSB (PICS Before) provides the correct signal conditioning to allow a volt-free contact type touch probe to be used.

Input PICSA (PICS After) allows an externally conditioned probe signal to be used.

The probe function works like the marker function but latches all three axis counters simultaneously. Unlike the marker function it operates asynchronously and hence does not require that the A and B signals are at a logic one level.

5.1.3.1 Probe input PICSB

The Deva004 implements the Renishaw PICSB interface for systems that do not include a probe signal conditioning module. Probe signal conditioning is implemented on the Deva004.

The probe input is suitable for use with Renishaw touch trigger probes. The input is active low and is terminated in a 27k pull-up resistor to 10v. The signal conditioning circuits operate from a stabilised 10v supply and follows Renishaws guidelines.

Other signals implemented are: STOP, PPOFF, PLED, PDAMP, LEDOFF.

The card also incorporates an output to drive the probe led.

5.1.3.2 Probe input PICSA

The Deva004 implements the Renishaw PICSA interface for systems that do include a probe signal conditioning module.

The following signals are implemented: STOP, PPOFF, ERROR, SYNC, HALT, PDAMP, LEDOFF, READ.

5.1.4 Digital IO

The DEVA004 has 48 digital IO lines arranged in 3 groups of 16 bits. The first group provides 16 TTL level I/Os and may be associated with special functions specific to other features of the card. The second and third group of IO lines provide 16 opto-isolated digital inputs and 16 opto-isolated digital outputs, respectively.

5.1.5 Pulse Generator

The DEVA004 has a 16 bit Pulse Generator circuit which can produce a pulse every n counts of the X axis encoder, where n is programmable from 0 to 65535. The pulse can be sent directly to a digital output, cause a hardware event or generate an interrupt. Please refer to the event system diagrams for details of which actions can be triggered by the pulse generator event.

The pulse generator may be started and stopped through software or hardware control. If hardware control is selected the axis compare functions are used as start and stop positions.

The pulse generator has both uni-directional and bi-directional modes, configurable direction when in uni-directional mode and incorporates an optional de-glitch circuit.

5.1.6 Axis Compare

The X axis has two axis compare registers which may be used to generate events when the axis position matches the compare position. The axis compare registers are shared with the Marker and Probe latches.

5.1.7 Timer 0

Timer 0 is an 8 bit interval timer which can be programmed in steps of 0.1ms up to a maximum of 25.6ms. This timer may be used to generate interrupts, latch encoder positions, trigger a DAC write operation or trigger a digital output for external circuitry synchronisation.

5.1.8 Timer 1

Timer 1 is an 4 bit interval timer which can be programmed in steps of 0.1ms up to a maximum of 1.6ms. This timer may be used to trigger a DAC write operation or trigger a digital output for external circuitry synchronisation. Timer 1 can be chained with Timer 0 to allow for an accurate time interval between reading encoder positions and writing to the Digital to Analogue Converters.

5.1.9 Time Stamper

The time stamper is a 16 bit high speed timer with a resolution of 1us. It can be programmed to latch its value on an event so to provide an accurate time stamp at which the event occurred.

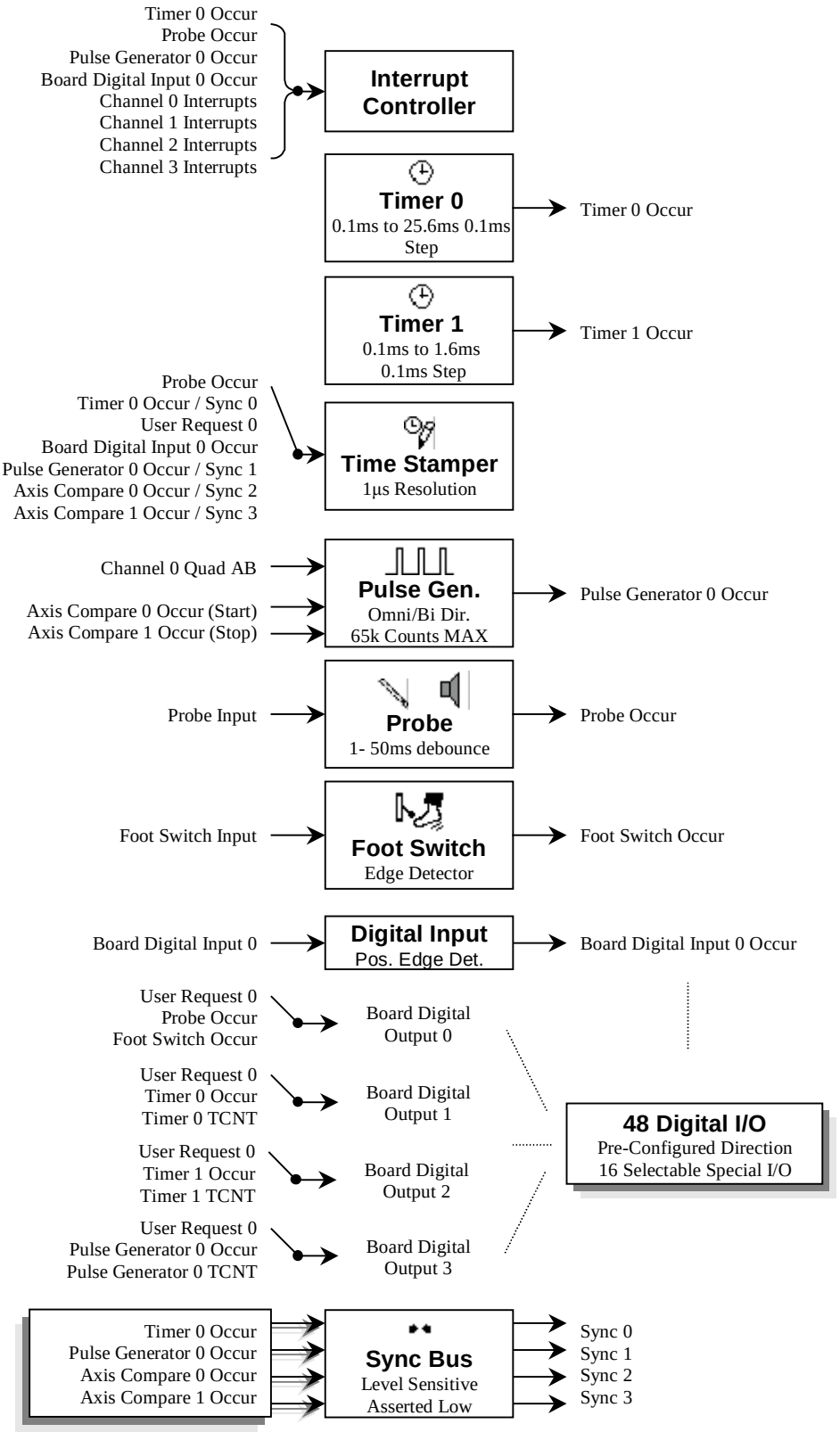
5.1.10 Event System

Many features of the card produce events. When an event occurs a flag called the occur flag is set. The occur flag may be polled by the user to establish that the event has occurred.

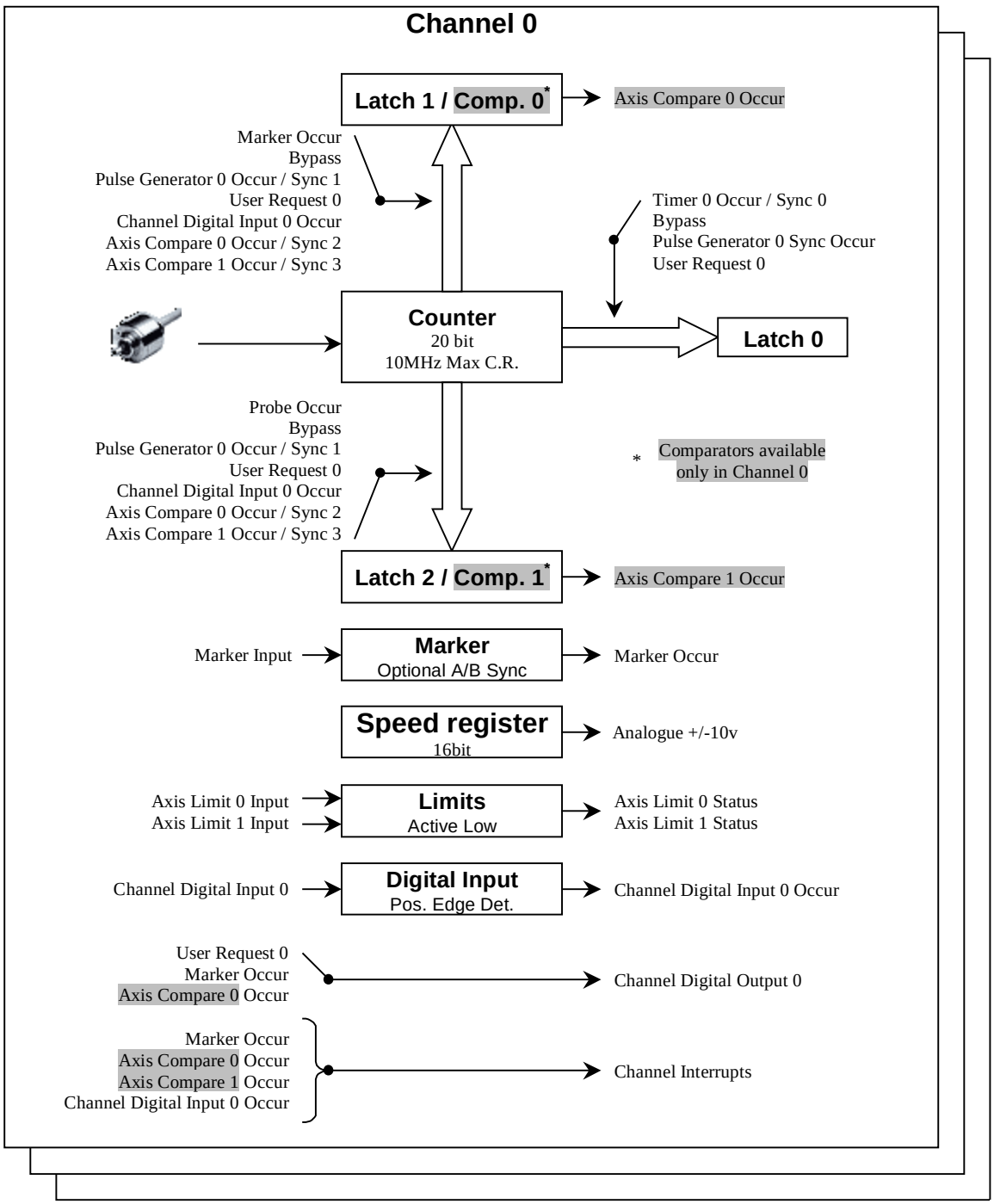
The card has a variety of actions which may be configured to respond to events. The actions include latching the current position in one of three latches, generating an interrupt, triggering a digital output or latching a time stamp of the event.

The following diagrams show the board level and channel level events and actions which may be programmed.

5.1.10.1 Incremental board level block diagram

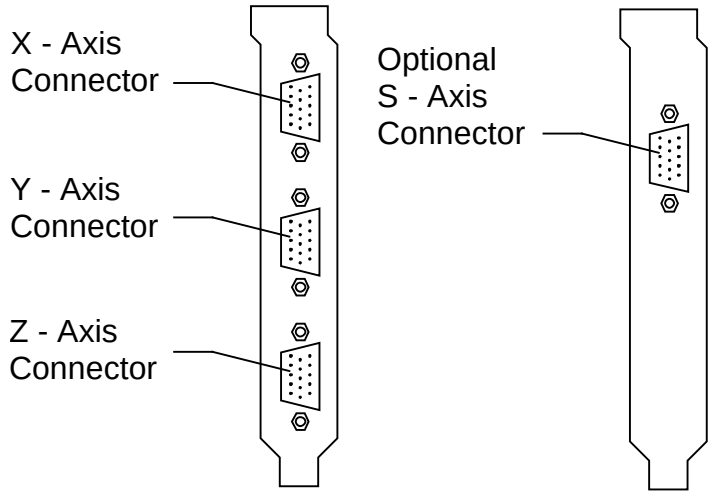


5.1.10.2 Channel level block diagram



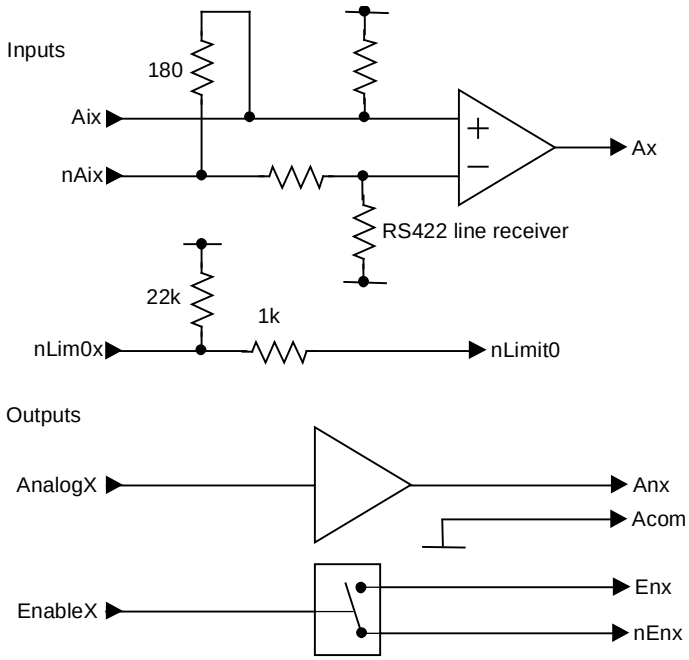
5.2 Axis connection details

The DEVA004 has three axis connectors on the main card and an optional fourth channel connector on an additional bracket. Connection of each channel to the outside world is made through a 15 way D-type connector. The designation of the connectors is shown in the diagram below.



Connections to these sockets should be made with reference to the following pin-out table and simplified input circuit.

Pin Number (15 way HD female 'D')	Signal	Function
1	Ai	A phase input
2	Bi	B phase input
3	En	Motor enable
4	Mi	Marker input
5	nLim0	nLimit 0
6	nAi	nA phase input
7	nBi	nB phase input
8	nEn	nMotor enable
9	nMi	nMarker input
10	nLim1	nLimit 1
11	+12V	+12 volts supply
12	+5V	+5 volt supply
13	0V	0 volt common
14	An	Analog output
15	Acom	Analog common



Note: Do not connect the 15-way D-type plug from a VGA monitor into one of the encoder input channels, as damage may result.

5.2.1 Axis signal descriptions

The Ai & nAi, Bi & nBi inputs are differential pairs for connection to the A-phase and B-phase quadrature outputs of an incremental encoder.

The Mi & nMi inputs are differential inputs for the channel marker signal. This function will latch the counter reading for the relevant channel, allowing an accurate reading of the position of a moving encoder to be made at a specific instant. The marker function does not stop the counter itself, which is able to continue reading the encoder position and so it will not cause the card to lose track of the system's datum position.

The inputs nLim0 and nLim1 are digital inputs intended to be used to connect normally closed overtravel inputs. Note it is also possible to configure the software to use one or more of the opto-isolated inputs to achieve this function.

The differential inputs use RS422 levels which accept OV for logic low, and from +5V to +12V for logic high. Because they are differential inputs, one input should be low when the other is high. For Example, to trigger the marker function, set Mi input to high, and set nMi input to low. To turn the marker function off, reverse these voltages.

An internal resistor network is provided which allows connection of single ended signals to the non-inverting inputs.

The analog output is +/- 10v single ended with reference to the analog common signal. Analog common and 0v are connected internally.

The nEn and En connections are an opto-isolated MOSFET volt free (SSR) pair which are short circuited when the drive is enabled.

Note the Analog and Enable signals are also provided on the IO2 internal servo connector.

The +12V and +5V power supply pins may be used to supply the dc power requirements for the encoders.

5.2.2 Renishaw probe input connections

The Renishaw connection kit provides a 9 way female 'D' connector which may be used with the bracket supplied or mounted in a free cut-out on the PC case. This kit should be connected to either the PICSA or PICSB header on the main board. Connections to this socket should be made with reference to the following pin-out table.

Pin Number (9 way female 'D')	**Pin Number (Header)**	**PICS A**	**PICSB**
1	1	STOP	STOP
2	3	PPOFF	PPOFF
3	5	0v	0v
4	7	ERROR	PLED A
5	9	SYNC	PRB
6	2	HALT	+5v
7	4	PDAMP	PDAMP
8	6	LEDOFF	LEDOFF
9	8	READ	PRB RET
-	10	0v	0v

5.2.3 Digital / Analog IO connections

There are three headers labelled Feature, IO1 and IO2 which provide access to a variety of digital and analog IO.

5.2.3.1 Feature connector

The Feature IO conforms to 5v TTL levels, Voh min 2.4v at –8ma and Vil maximum 0.4 at 12ma. The IO1 and IO2 connectors.

Pin (Header)	Connector IO0 (FEATURE)	Special Function
1	IO0 / UR0 / MOx / AC0Ox	User request 0 / Marker occur x / Axis compare 0 occur
2	IO1 / UR0 / MOy	User request 0 / Marker occur y
3	IO2 / UR0 / MOz	User request 0 / Marker occur z
4	IO3 / UR0 / MOs	User request 0 / Marker occur s
5	IO4 / UR0 / PO / FO	User request 0 / Probe occur / Footswitch occur
6	IO5 / UR0 / TMRO / TMRT0CNT	User request 0 / Timer 0 occur / Timer 0 terminal count
7	IO6 / UR0 / TMR1 / TMRT1CNT	User request 0 / Timer 1 occur / Timer 1 terminal count
8	IO7 / UR0 / PG0O / PG0TCNT	User request 0 / Pulse gen. 0 occur / Pulse gen. 0 terminal count
9	IO8 / CDI0x	Channel x digital input 0
10	IO9 / CDI0y	Channel y digital input 0
11	IO10 / CDI0z	Channel z digital input 0
12	IO11 / CDI0s	Channel s digital input 0
13	IO12 / BDI0	Board digital input 0
14	IO13	
15	IO14	
16	IO15	
17	GND	
18	GND	
19	+5v	
20	+12v	

5.2.3.2 IO1 Digital IO connector

This connector has 12 opto-isolated digital inputs sharing a digital input common connection that may be connected to either a positive or negative supply rail. For example, if the digital input common is connected to 0 volt, then each input reads a logic '1' when the input is supplied with 12-24 volt.

This connector also has 9 opto-isolated MOSFET volt free (SSR) outputs sharing a digital output common connection that may be connected to either a positive or negative supply rail. For example, if the digital output common is connected to 12-24 volt then the outputs will supply 12-24 volt to the load when a logic 1 is written to the output.

The watchdog output is an opto-isolated MOSFET volt free (SSR), the contacts are closed when the system is in the healthy state.

Pin Number (Header)	**Pin Number (25 way female 'D')**	**Description**
1	1	IO16 / Input 0
2	14	IO17 / Input 1
3	2	IO18 / Input 2
4	15	IO19 / Input 3
5	3	IO20 / Input 4
6	16	IO21 / Input 5
7	4	IO22 / Input 6
8	17	IO23 / Input 7
9	5	IO24 / Input 8
10	18	IO25 / Input 9
11	6	IO26 / Input 10
12	19	IO27 / Input 11
13	7	Digital Input Common
14	20	IO32 / Output 0
15	8	IO33 / Output 1
16	21	IO34 / Output 2
17	9	IO35 / Output 3
18	22	IO36 / Output 4
19	10	IO37 / Output 5
20	23	IO38 / Output 6
21	11	IO39 / Output 7
22	24	IO40 / Output 8
23	12	Digital Output Common
24	25	Watchdog+
25	13	Watchdog-
26	N.C.	GND

5.2.3.3 IO2 Internal servo connector

This connector has 4 opto-isolated digital inputs sharing a digital input common connection that may be connected to either a positive or negative supply rail. For example, if the digital input common is connected to 0 volt, then each input reads a logic '1' when the input is supplied with 12-24 volt.

This connector also has 3 opto-isolated MOSFET volt free (SSR) outputs sharing a digital output common connection that may be connected to either a positive or negative supply rail. For example, if the digital output common is connected to 12-24 volt then the outputs will supply 12-24 volt to the load when a logic 1 is written to the output.

An opto-isolated MOSFET volt free (SSR) axis enable signal and a single ended +/- 10v analog output is provided for each axis.

Pin Number (Header)	**Pin Number (25 way female 'D')**	**Description**
1	1	IO28 / Input 12
2	14	IO29 / Input 13
3	2	IO30 / Input 14
4	15	IO31 / Input 15
5	3	Digital Input Common
6	16	IO41 / Output 9
7	4	IO42 / Output 10
8	17	IO43 / Output 11
9	5	IO44 / EnableX +
10	18	IO44 / EnableX -
11	6	IO45 / EnableY +
12	19	IO45 / EnableY -
13	7	IO46 / EnableZ+
14	20	IO46 / EnableZ-
15	8	IO47 / EnableS+
16	21	IO47 / EnableS-
17	9	Analog Output Common
18	22	Analog Output X
19	10	Analog Output Common
20	23	Analog Output Y
21	11	Analog Output Common
22	24	Analog Output Z
23	12	Analog Output Common
24	25	Analog Output S
25	13	Digital Output Common
26	N.C.	GND

5.2.4 Sync connector

The sync connector provides a method of linking cards in hardware to allow events to be routed from one card to another. In general a ribbon cable connecting all 10 connections between all cards is required. Each of the bi-directional sync signals is triggered by a specific event and may be programmed to cause one or more actions.

Pin Number (Header)	Function	Events	Actions
1	Sync0	Timer 0 occur	Latch 0
2	GND		
3	Sync1	Pulse Generator 0 occur	Latch 0,1,2
4	GND		
5	Sync2	Axis Comparator 0 occur	Latch 1,2
6	GND		
7	Sync3	Axis Comparator 1 occur	Latch 1,2
8	GND		
9	GND		
10	GND		

5.3 Direct hardware programming

The DEVA004 incremental encoder interface card is supplied with a variety of device drivers for Microsoft Windows operating systems which perform all low level access functions required for its operation. Using a device driver offers several benefits including not having to read any more of this manual and the ability to re-use application software and routines with any of DEVA Electronics compatible products. The device drivers and the common software interface are described in section 3. For applications where hardware access is essential please contact Deva for further information

6 PCI Stepper control hardware

6.1 Functional description

The DEVA004 stepper control card is designed to control 3 or 4 stepper motors vi the PC PCI bus. The following sections describe the various functions of the interface. For a detailed description of how to program these facilities, please refer to section 3.

6.1.1 Stepper outputs

The DEVA004 provides Step and Direction signals for each axis along with a volt free contact used for motor enable or hold current selection.

6.1.2 Digital IO

The DEVA004 has 48 digital IO lines arranged in 3 groups of 16 bits. The first group provides 16 TTL level I/Os and may be associated with special functions specific to other features of the card. The second and third group of IO lines provide 16 opto-isolated digital inputs and 16 opto-isolated digital outputs, respectively.

6.1.3 Timer 0

Timer 0 is an 8 bit interval timer which can be programmed in steps of 0.1ms up to a maximum of 25.6ms. This timer may be used to generate interrupts, latch encoder positions or trigger a digital output for external circuitry synchronisation.

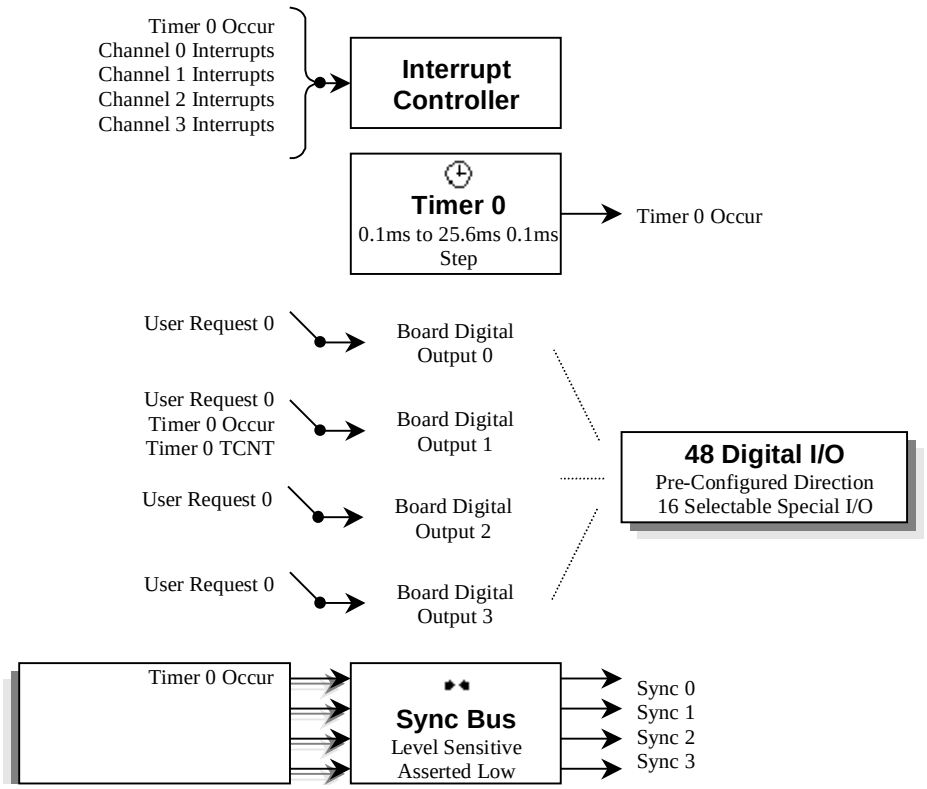
6.1.4 Event System

Many features of the card produce events. When an event occurs a flag called the occur flag is set. The occur flag may be polled by the user to establish that the event has occurred.

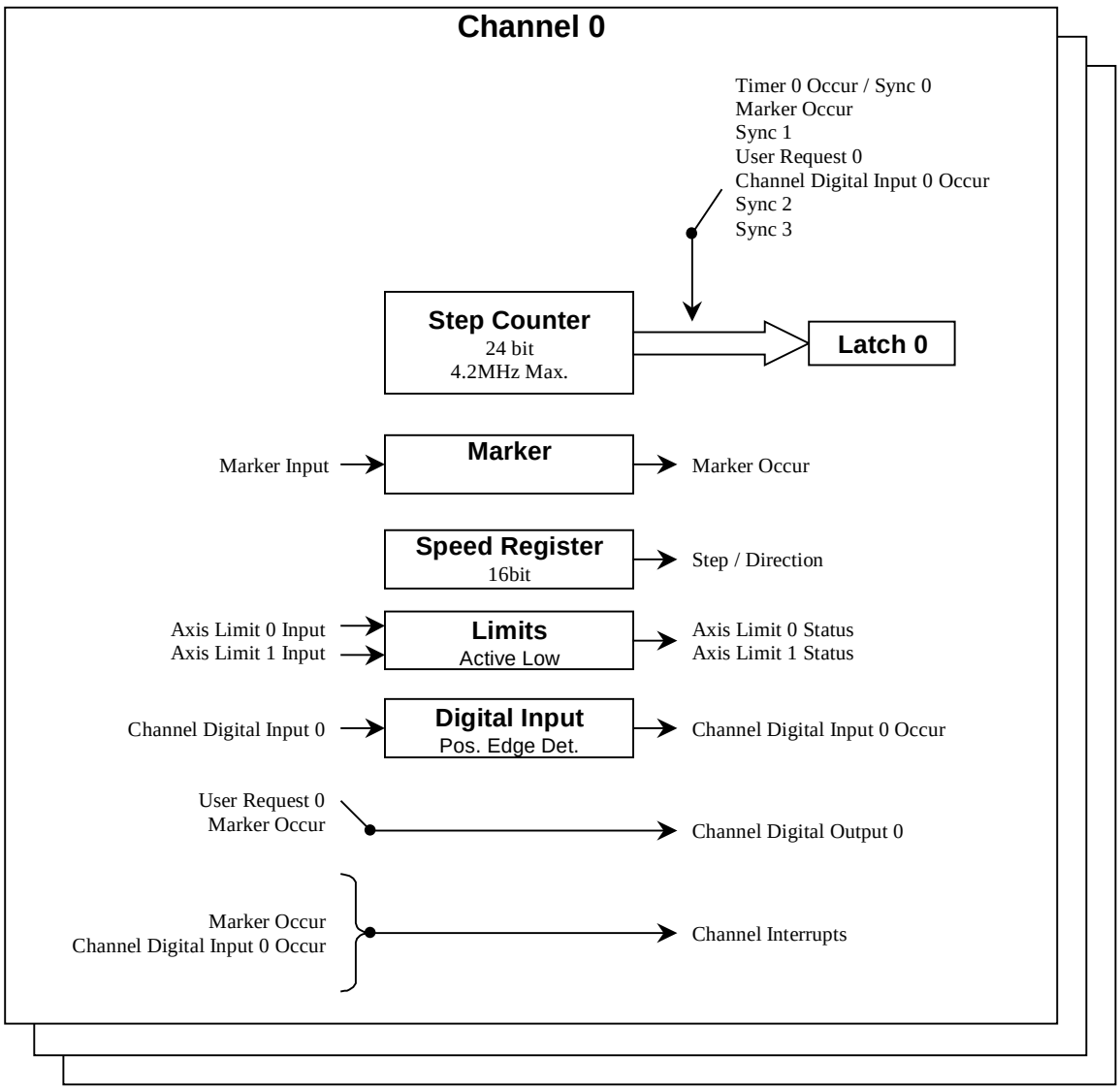
The card has a variety of actions which may be configured to respond to events. The actions include latching the current position in one of three latches, generating an interrupt, triggering a digital output or latching a time stamp of the event.

The following diagrams show the board level and channel level events and actions which may be programmed.

6.1.4.1 Board level block diagram

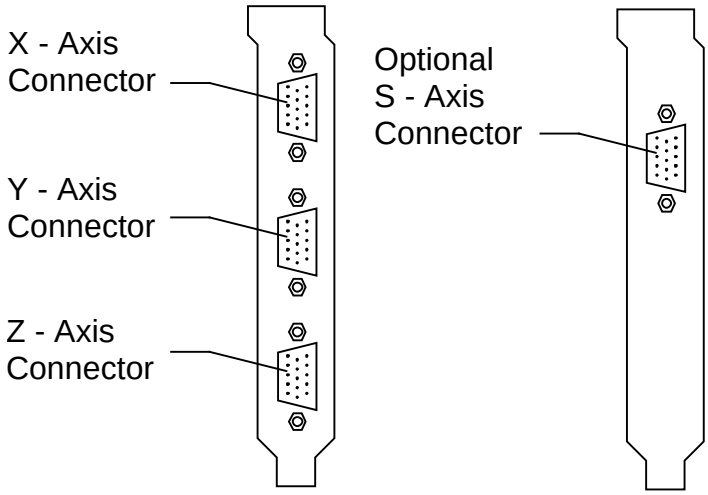


6.1.4.2 Channel level block diagram



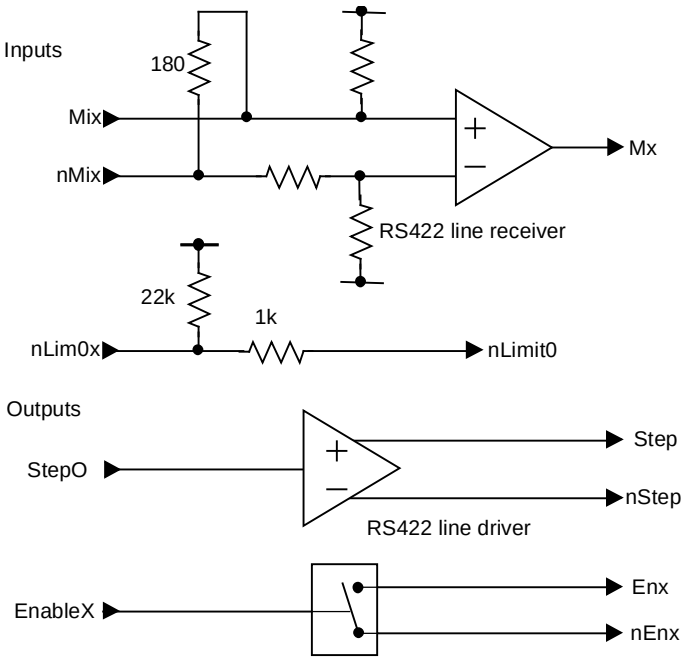
6.2 Axis connection details

The DEVA004 has three axis connectors on the main card and an optional fourth channel connector on an additional bracket. Connection of each channel to the outside world is made through a 15 way D-type connector. The designation of the connectors is shown in the diagram below.



Connections to these sockets should be made with reference to the following pin-out table and simplified input circuit.

Pin Number (15 way HD female 'D')	Signal	Function
1	Step	Step output
2	Dir	Dir output
3	En	Motor enable
4	Mi	Marker input
5	nLim0	nLimit 0
6	nStep	nStep output
7	nDir	nDir output
8	nEn	nMotor enable
9	nMi	nMarker input
10	nLim1	nLimit 1
11	+12V	+12 volts supply
12	+5V	+5 volt supply
13	0V	0 volt common
14	n.c.	Do not use
15	0V	Do not use



Note: Do not connect the 15-way D-type plug from a VGA monitor into one of the encoder input channels, as damage may result.

6.2.1 Axis signal descriptions

The Step signal outputs a positive pulse for each required step of the motor. The Dir output indicates the required direction. The inverted outputs nStep and nDir are available if required.

The inputs nLim0 and nLim1 are digital inputs intended to be used to connect normally closed overtravel inputs. Note it is also possible to configure the software to use one or more of the opto-isolated inputs to achieve this function.

The Mi & nMi inputs are differential inputs for the channel marker signal

The differential inputs use RS422 levels which accept OV for logic low, and from +5V to +12V for logic high. Because they are differential inputs, one input should be low when the other is high. For Example, to trigger the marker function, set Mi input to high, and set nMi input to low. To turn the marker function off, reverse these voltages.

An internal resistor network is provided which allows connection of single ended signals to the non-inverting inputs.

The nEn and En connections are an opto-isolated MOSFET volt free (SSR) pair which are short circuited when the drive is enabled.

Note the Enable signals are also provided on the IO2 internal servo connector.

The +12V and +5V power supply pins may be used to supply the dc power requirements for the encoders.

6.2.2 Digital IO connections

There are three headers labelled Feature, IO1 and IO2 which provide access to a variety of digital IO.

6.2.2.1 Feature connector

The Feature IO conforms to 5v TTL levels, Voh min 2.4v at –8ma and Vil maximum 0.4 at 12ma. The IO1 and IO2 connectors.

Pin (Header)	Connector IO0 (FEATURE)	Special Function
1	IO0 / UR0 / MOx	User request 0 / Marker occur x
2	IO1 / UR0 / MOy	User request 0 / Marker occur y
3	IO2 / UR0 / MOz	User request 0 / Marker occur z
4	IO3 / UR0 / MOs	User request 0 / Marker occur s
5	IO4 / UR0	User request 0
6	IO5 / UR0 / TMRO / TMRT0CNT	User request 0 / Timer 0 occur / Timer 0 terminal count
7	IO6 / UR0	User request 0
8	IO7 / UR0	User request 0
9	IO8 / CDI0x	Channel x digital input 0
10	IO9 / CDI0y	Channel y digital input 0
11	IO10 / CDI0z	Channel z digital input 0
12	IO11 / CDI0s	Channel s digital input 0
13	IO12	
14	IO13	
15	IO14	
16	IO15	
17	GND	
18	GND	
19	+5v	
20	+12v	

6.2.2.2 IO1 Digital IO connector

This connector has 12 opto-isolated digital inputs sharing a digital input common connection that may be connected to either a positive or negative supply rail. For example, if the digital input common is connected to 0 volt, then each input reads a logic '1' when the input is supplied with 12-24 volt.

This connector also has 9 opto-isolated MOSFET volt free (SSR) outputs sharing a digital output common connection that may be connected to either a positive or negative supply rail. For example, if the digital output common is connected to 12-24 volt then the outputs will supply 12-24 volt to the load when a logic 1 is written to the output.

The watchdog output is an opto-isolated MOSFET volt free (SSR), the contacts are closed when the system is in the healthy state.

Pin Number (Header)	Pin Number (25 way female 'D')	Description
1	1	IO16 / Input 0
2	14	IO17 / Input 1
3	2	IO18 / Input 2
4	15	IO19 / Input 3
5	3	IO20 / Input 4
6	16	IO21 / Input 5
7	4	IO22 / Input 6
8	17	IO23 / Input 7
9	5	IO24 / Input 8
10	18	IO25 / Input 9
11	6	IO26 / Input 10
12	19	IO27 / Input 11
13	7	Digital Input Common
14	20	IO32 / Output 0
15	8	IO33 / Output 1
16	21	IO34 / Output 2
17	9	IO35 / Output 3
18	22	IO36 / Output 4
19	10	IO37 / Output 5
20	23	IO38 / Output 6
21	11	IO39 / Output 7
22	24	IO40 / Output 8
23	12	Digital Output Common
24	25	Watchdog+
25	13	Watchdog-
26	N.C.	GND

6.2.2.3 IO2 Internal servo connector

This connector has 4 opto-isolated digital inputs sharing a digital input common connection that may be connected to either a positive or negative supply rail. For example, if the digital input common is connected to 0 volt, then each input reads a logic '1' when the input is supplied with 12-24 volt.

This connector also has 3 opto-isolated MOSFET volt free (SSR) outputs sharing a digital output common connection that may be connected to either a positive or negative supply rail. For example, if the digital output common is connected to 12-24 volt then the outputs will supply 12-24 volt to the load when a logic 1 is written to the output.

An opto-isolated MOSFET volt free (SSR) axis enable signal is provided for each axis.

Pin Number (Header)	Pin Number (25 way female 'D')	Description
1	1	IO28 / Input 12
2	14	IO29 / Input 13
3	2	IO30 / Input 14
4	15	IO31 / Input 15
5	3	Digital Input Common
6	16	IO41 / Output 9
7	4	IO42 / Output 10
8	17	IO43 / Output 11
9	5	IO44 / EnableX +
10	18	IO44 / EnableX -
11	6	IO45 / EnableY +
12	19	IO45 / EnableY -
13	7	IO46 / EnableZ+
14	20	IO46 / EnableZ-
15	8	IO47 / EnableS+
16	21	IO47 / EnableS-
17	9	GND
18	22	n.c.
19	10	GND
20	23	n.c.
21	11	GND
22	24	n.c.
23	12	GND
24	25	n.c.
25	13	Digital Output Common
26	N.C.	GND

6.2.3 Sync connector

The sync connector provides a method of linking cards in hardware to allow events to be routed from one card to another. In general a ribbon cable connecting all 10 connections between all cards is required. Each of the bi-directional sync signals is triggered by a specific event and may be programmed to cause one or more actions.

Pin Number (Header)	Function	Events	Actions
1	Sync0	Timer 0 occur	Latch 0
2	GND		
3	Sync1		Latch 0
4	GND		
5	Sync2		Latch 0
6	GND		
7	Sync3		Latch 0
8	GND		
9	GND		
10	GND		

6.3 Direct hardware programming

The DEVA004 incremental encoder interface card is supplied with a variety of device drivers for Microsoft Windows operating systems which perform all low level access functions required for its operation. Using a device driver offers several benefits including not having to read any more of this manual and the ability to re-use application software and routines with any of DEVA Electronics compatible products. The device drivers and the common software interface are described in section 3. For applications where hardware access is essential please contact Deva for further information.

52 Woodside Business Park
Birkenhead
Wirral CH41 1EL
United Kingdom

Tel +44 (0)151 647 3222
Fax +44 (0)151 647 4511

Email: support@deva.co.uk

Web : www.deva.co.uk